

FLASH TOIMINTAPELITOTEUTUKSESSA
CASE: NATO COMMANDO

Ville Lähde

Tampereen ammattikorkeakoulu
Viestinnän osaston lopputyö
Vuorovaikutteisuuden suunnittelu
2005
Ville Lähde

OPINNÄYTETIIIVISTELMÄ

Osasto Viestintä	Erikoistumisala Vuorovaikutteisuuden suunnittelu
Tekijä Ville Lähde	
Työn nimi: Flash toimintapelitoteutuksessa	
Lopputyön laji Mediateko	
Työn valmistumisaika 18.4.2005	Sivumäärä 44
<p>Tiivistelmä</p> <p>Tässä työssä tutkitaan, voiko Macromedian Flashillä toteuttaa internetissä toimivan toimintapelin. Case on peli nimeltä Nato Commando.</p> <p>Työssä käydään läpi mikä on online-peli, pelinteon työkalut, sekä pelinteon eri vaiheet. Viimeisessä kappaleessa analysoidaan, kuinka tavoitteessa onnistuttiin ja oliko projekti muuten onnistunut.</p> <p>This work describes how one can create a workable online action game with Macromedia Flash. The case is an action game called Nato Commando.</p> <p>The work describes what is an online action game, the tools for game programming and the different stages of game developing. In the final chapter I analyze how well the goal was achieved and how successful was the entire project.</p>	
Aineisto Meigs, Tom: Ultimate Game Design, McGraw-Hill/Osborne 2003	
Besley, Bhargal, Eden, Ferguson, Monnone, Peters, Rhodes, Young: Flash Mx 2004 Games Most Wanted, Springer-Verlag 2004	
Makar, Jobe: Macromedia Flash MX Game Design Demystified, Macromedia Press, 2003	
Asiasanat Flash, peli, internet, actionScript, xml	
Säilytyspaikka TTVO kirjasto	
Muita tietoja	

Sisällys

1. JOHDANTO	5
1.1. Mikä on online-toimintapeli?	5
1.2. Pelin esittely	6
2. ONLINE-PELIN TYÖKALUT	7
2.1. Macromedia Flash	7
2.2. Työkalu: xml	9
2.3. Työkalu: PHP	10
2.4. Muita työkaluja	10
2.4.1. Java	10
2.4.2. JavaScript	11
2.4.3. Macromedia Director	11
3. PELIN TOTETUS	12
3.1. projekti	12
3.2. Suunnitteluvaihe	12
3.2.1. Pelihahmon liikuttaminen	13
3.2.2. Törmäystarkastelu	14
3.2.3. Viholliset	14
3.2.4. Pelimaailma	14
3.3. Ohjelmointi	15
3.3.1. Pelihahmon kontrollointi	15
3.3.2. Pelimaailman kuvaaminen xml-tiedostoilla	16
3.3.3. Pelimaailman luonti ja liikkuminen ruutujen välillä	17
3.3.4. Törmäystarkastelu ja interaktiiviset esineet	19
3.3.5. Vastustajat	20
3.3.6. Miinat	21
3.3.7. Luodit	21
3.4. Pelikenttien rakentaminen	22
3.5. Pelitestaus	24

4. LOPPUTULOS	26
4.1. Hyvät puolet	26
4.1.1. Pelimoottori	26
4.1.2. Keinoäly	26
4.1.3. Kenttäeditori	27
4.1.4. xml	27
4.2. Huonot puolet	27
4.2.1. Grafiikka	27
4.2.2. Vihollisten määrä	28
4.2.3. Optimointi	28
4.3. projektista	28
4.4. Lisäideoita peliin	29
4.4.1. Aikarajoitus	29
4.4.2. Panttivangit ja muut ei-vihamieliset hahmot	29
4.4.3. Enemmän lisäaseita	
30	
4.4.4. Kulkuneuvon käyttäminen	30
4.4.5. Elämät	
30	
4.4.6. EventTilet	30
4.4.7. Vihollispesäkkeet	31
4.4.8. Lisää vaarallisia tilejä	31
4.4.9. Teleportit ja liikuteltavat sekä tuhoutuvat tilet	31
4.5. Lopuksi	31
LÄHTEET	33
LIITTEET	34

1. JOHDANTO

Tässä lopputyössä kuvataan Macromedia Flash-pohjaisen toimintapelin suunnittelua, kehittämistä ja ohjelmointia. Tarkoituksena on tutkia, voiko Flashillä toteuttaa netissä toimivan toimintapelin. Käytän esimerkkinä työryhmäni kanssa toteuttamaani peliä nimeltä Nato Commando. Samalla selvitetään Online-toimintapeliin taustoja ja muita pelien toteuttamistapoja. Loppukappaleessa pohditaan, saavutettiin tavoite, mitä olisi voinut tehdä paremmin ja olisiko joku muu toteuttamistapa soveltunut tähän caseen paremmin. Lukijalta oletetaan peruskäsitteiden ymmärrystä internetin toiminnasta ja HTML-kielestä, sekä peliohjelmoinnin peruskäsitteistä.

1.1 Mikä on online-peli?

Online-pelit, eli Internet-pelit ovat pelejä, joita pelataan internetin välityksellä. Nämä pelit eroavat perinteisistä tietokone- ja videopeleistä siten, että ne ovat laiteriippumattomia. Tavallisesti Internet-peli vaatii toimiakseen internet-selaimen ja lisäosan (plug-in), jolla peli on suunniteltu toimimaan. (Wikipedia, The Free Encyclopedia, 2004, saatavilla www-muodossa: http://en.wikipedia.org/wiki/Online_game (luettu 12. 4. 2005))

Internet-peli on useimmiten client side -pohjaisia, mikä tarkoittaa että peli ladataan pelaajan koneelle ja suoritetaan siellä. Myös server side -pohjaisia sovelluksiakin löytyy. Näissä pelin suoritus ajetaan palvelinkoneella. Server side -pohjaiset pelit ovat usein vuoropohjaisia strategiapelejä, eivätkä ne ole graafisesti yhtä näyttäviä kuin client side -pohjaiset pelit. (Wikipedia, The Free Encyclopedia, 2004, saatavilla www-muodossa: http://en.wikipedia.org/wiki/Online_game (luettu 12 .4. 2005))

(Ks. liite 1.)

1.2. Pelin esittely

Lopputyönä tekemäni mediateko on Nato Commando -niminen nettipeli. Peli on toteutettu Macromedian Flash -ympäristössä ja siinä käytetään hyväksi xml-tiedostoja peliruutujen ja highscore-listan luomiseen. Idea peliin tuli filmiryhmäni Trash Videon parhaillaan tekeillä olevasta samannimisestä toimintaelokuvasta. Pelin tarkoituksena on paitsi tarjota viihdettä pelaajalleen, mutta myös levittää sanaa syksyllä 2005 ilmestyvästä Nato Commando -elokuvasta.

Pelin juoni seuraa löyhästi elokuvan tapahtumia: Naton kovin kommando, Alex Lommo, lähtee selvittämään oppi-isänsä Everstiluutnantti Fägerlundin mysteeristä kidnappaamista. Jäljet johtavat Metacom -nimisen konsernin jäljille. Tästä seuraa loputon määrä ammuskelua ja verisiä taistelukohtauksia. Mukana on myös UFO -olioiden salakavala juoni maailman valtaamiseksi.

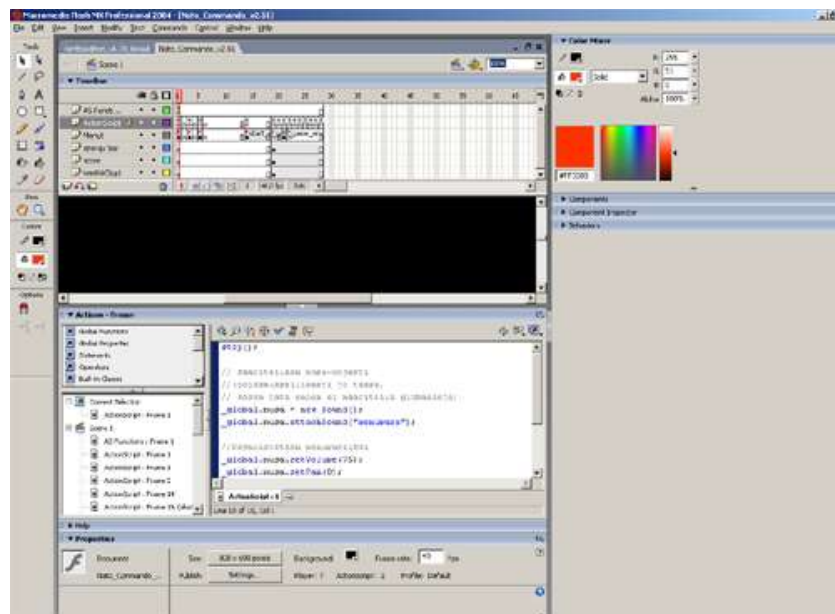
Peli on lintuperspektiivistä kuvattu toimintaseikkailu. Pelihahmoa ohjataan näppäimistön nuolinäppäimillä ja ctrl-näppäimellä ammutaan. Pelaajan vastustajina on terroristeja ja Metacom -miehiä. Pelin henki on selvä, tapa tai tule tapetuksi. Ammuttuaan ruudun tyhjäksi vihollisista, pelaaja voi suunnata seuraavaan ruutuun kävelemällä peliruudun ulkopuolelle. Uusi peliruutu latautuu ja ohjelma arpoo xml-tiedoston pohjalta vastustajien määrän ja tyyppin. Pelin tarkoituksena on päästä Metacom -kompleksin sydämeen ja tuhota päävihollinen. Matkan varrella löytyy erilaisia bonusesineitä: rahasäkki, disketti DNA-kapseli. Pelialueelta löytyy myös tehokkaampia lisäaseita vihollisten nujertamiseksi ja avaimia, joita on kerättävä päästäkseen eteenpäin ruuduista joissa on lukittu ovi.

2. ONLINE-PELIN TYÖKALUT

Seuraavassa esittelen muutamia työkaluja, joita on käytetty Nato Commando -projektissa.

2.1. Macromedia Flash

Macromedian kehittämä Flash tarjoaa kehittäjille mahdollisuuden julkaista netissä vektorigrafiikkaa, animaatiota, ääntä ja videota alhaisen kaistatilan asettamien rajojen puitteissa. Flashiin on sisäänrakennettu oma oliopohjainen ohjelmointikieli, actionScript, jolla saa ohjelmoitua monimutkaisiakin toimintoja. Flash pitää sisällään ikkunoista koostuvan graafisen työympäristön. Ohjelman päätyökaluja ovat Stage, esityksen näkymä, Timeline, esityksen aikajana, Library, joka pitää sisällään kaikki esityksen elementit, sekä Actions joka toimii koodi-ikkunana. (Wikipedia, The Free Encyclopedia, 2004, saatavilla www-muodossa: http://en.wikipedia.org/wiki/Macromedia_Flash (luettu 13 .4. 2005))



1. Näkymä Flash-käyttöliittymästä

Flash -esityksessä voidaan käyttää lähes kaikkia tunnetuimpia mediaelementtejä ja tiedostomuotoja. Ääni-, video-, kuva-, ja

animaatiotiedostot tuodaan ohjelman käytettäväksi sisään ohjelman kirjastoon. Jokaisella auki olevalla Flash -projektilla on oma kirjastonsa. Käyttäjä voi luoda kirjastoon kansioita, joilla voi paremmin hallita projektin tiedostotulvaa. Ääni ja video ovat eniten kaistaa ja palvelintilaa syöviä elementtejä. Niitä sisään tuodessaan Flashiin käyttäjällä on erilaisia pakkaustyökaluja tiedoston pakkaamiseksi kompaktimpaan tilaan. (Macromedia, Using flash, 2004)

Flash tallentaa omat työtiedostonsa fla -päätteisinä. Nämä tiedostot pitävät sisällään kaiken tiedon, mitä projektin kirjastoon on tuotu sisälle, paitsi ne tiedostot, joita käytetään ulkoisesta kirjastosta. Tästä johtuen fla -tiedostot saattavat paisua varsin isokokoisiksi. Fla -tiedostoja ei pysty sellaisenaan jakamaan käytettäväksi netissä, niiden käynnistäminen vaatii Flash -ohjelman. Jotta Flashillä toteutettu projekti saadaan selailtavaksi internetissä, on siitä tehtävä swf -päätteinen tiedosto. Näin selaimen asennettu lisäosa osaa näyttää flash-sisältöä internet-sivulla. swf-tiedostoja voi upottaa HTML-dokumenttiin aivan samalla tavalla kuin esim. jpg-kuvatiedostoja. (Macromedia, 2004, Using flash)

Flashin tukee lähes kaikkia mahdollisia graafisia elementtejä. Grafiikkaa voi tuoda sisään Flashiin pikselipohjaisena, mikä tarkoittaa sitä että kuva koostuu pienistä neliöistä. Yleisimpiä pikseliformaatteja ovat mm. GIF, JPEG, PNG ja BMP. Tavallisimpia ohjelmia tämänkaltaisten tiedostojen luomiseen ovat mm. Adobe Photoshop, Macromedia Fireworks ja Microsoft Paint. (Makar, Jobe: Macromedia Flash MX Game Design Demystified, sivu 274)

Vektorigrafiikalla tarkoitetaan puolestaan matemaattista kaaviota, jota käytetään kuvaamaan erilaisia muotoja, jotka muodostavat kuvan. Vektorigrafiikkaa voi tehdä ohjelmilla Macromedia Flash, Macromedia Freehand, Corel Draw ja Adobe Photoshop muutamia mainitakseni. Yleisimpiä vektorigrafiikkaformaatteja ovat EPS ja SWF. (Makar, Jobe: Macromedia Flash MX Game Design Demystified, sivu 274)

Pikseli- ja vektorigrafiikassa on molemmissa hyvät ja huonot puolensa. Pikseligrafiikkakuvat ovat korkealaatuisia ja tarkkoja, mutta ne saattavat viedä paljon tilaa ja niiden skaalaaminen isommaksi heikentää laatua. Vektorigrafiikka

puolestaan on erittäin vähän resursseja vievää ja loputtomasti skaalattavissa, mutta kovinkaan monimutkaisten kuvioiden luominen ei ole sillä järkevää. (Makar, Jobe: Macromedia Flash MX Game Design Demystified, sivu 274)

2.2. Työkalu: xml

Xml, eli Extensible Markup Language on World Wide Web (W3C) konsortion rekisteröimä ohjelmointikieli informaation tallentamiseen. Xml perustuu HTML kielestä tuttuihin tageihin. Erona HTML:ään on se, että käyttäjät voivat itse määritellä omat taginsa. (iDataExpress, internet glossary, 2005, saatavilla www-muodossa: <http://www.idataexpress.com/support/glossary.htm> (luettu 14.4.2005)).

Xml-tiedosto koostuu elementeistä, jotka voivat pitää sisällään lapsielementtejä. Näillä elementeillä voidaan kuvata monimutkaisia tietorakenteita. Seuraavassa esimerkki xml-tiedostosta, joka kuvaa yksinkertaista sähköpostiviestiä.

```
<mail>
  <lähettäjä>
    <nimi>
      Ville Lähde
    </nimi>
    <email>
      ville.lahde@cult.tpu.fi
    </email>
  </lähettäjä>
  <viesti>
    Morjensta!
  </viesti>
</mail>
```

Elementtejä voidaan yhdistää, jolloin niistä muodostuu nodeja ja koodia saadaan tiivistettyä.

```
<mail>  
  <lähettäjä nimi="Ville Lähde" email="ville.lahde@cult.tpu.fi" />  
  <viesti="Morjensta!" />  
</mail>
```

2.3. Työkalu: PHP

PHP on palvelinpuolen (server-side) ohjelmointikieli, joilla saadaan aikaiseksi mm. dynaamisia internet-sisältöjä (Terena, Guide to Network Resource Tools, 2005, saatavilla www-muodossa:

<http://www.terena.nl/library/gnrt/appendix/glossary.html> (luettu 14.4.2005)).

Nato Commandossa PHP:tä käytetään tallentamaan palvelimelle pistelistaus-xml ja editoriosuudessa kenttädatan sisällä pitävän xml-tiedosto.

2.4. Muita työkaluja

Seuraavassa on lyhyesti esiteltyinä muutamia muita keinoja luoda graafisia esityksiä, sekä interaktiivisia ohjelmia internettiin. Näitä ei kuitenkaan käytetty Nato Commandoa tehdessä.

2.4.1 Java

Java on Sun Microsystemsin kehittämä verkko-orientoitunut ohjelmointikieli. Sen avulla taidokas ohjelmoija pystyy luomaan vaativia internet-sovelluksia, kuten animaatiota, pelejä jne. Javan avulla voi tehdä kutakuinkin minkälaisen tietokoneohjelman tahansa ja sen voi upottaa internet sivulle samalla tavalla kuin Flashin SWF:n. (UC Berkeley, Glossary of Internet & Web Jargon, 2004, saatavilla www-muodossa:

<http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/Glossary.html> (luettu 28.2.2005)) Java vaatii toimiakseen selaimelle asennetun raskaan lisäosan.

Java-lisäosa on Flashin jälkeen suosituin selainten lisäpalikka 87.1 % selainpenetraatiollaan (Macromedia, Macromedia Flash Player Statistics, 2005, saatavilla www-muodossa:

http://www.macromedia.com/software/player_census/flashplayer/ (luettu

28.2.2005)).

2.4.2 JavaScript

JavaScript on Netscapen kehittämä simppele ohjelmointikieli internetiin, jonka tarkoituksena on parantaa nettisivujen interaktiivisuutta. Se toimii yhteistyössä html-kielen kanssa mahdollistaen dynaamiset toiminnot. JavaScriptissä on samoja ominaisuuksia kuin Javassa, mutta se on itsenäinen ohjelmointikieli. Java soveltuu parhaiten vuoropohjaisten pelien toteuttamiseen. (Eseehosting, Support glossary, 2005, saatavilla www-muodossa: <http://www.eseehosting.com/support/glossary.php/> (luettu 12.4.2005)).

2.4.3 Macromedia Director

Macromedia Director on Macromedian kehittäämä monipuolinen mediatyökalu. Se on alunperin suunniteltu tuottamaan CD- ja DVD sisältöjä. Directorissa käytetään Lingo-nimistä ohjelmointikieltä, mutta uusimman päivityksen yhteydessä siinä voidaan käyttää myös JavaScript -kieltä. Director tukee vektorigrafiikkaa ja 3D-animaatiota. Se integroituu hyvin Flash -animaation kanssa. Directorilla luotuja sisältöjä voidaan katsoa selaimella, mikäli ne tallennetaan shockwave -muotoisina. (Wikipedia, The Free Encyclopedia, 2004, saatavilla www-muodossa: http://en.wikipedia.org/wiki/Macromedia_Director (luettu 12 .4. 2005))

Shockwave vaatii oman lisäosansa (2468 kt, versio 10). Lisäosan selainpenetraatio on huomattavasti Flashiä ja Javaa pienempi, vain 52 prosenttia selainkannasta (Macromedia, Macromedia Flash Player Statistics, 2005, saatavilla www-muodossa: http://www.macromedia.com/software/player_census/shockwaveplayer/ (luettu 28.2.2005)).

3. PELIN TOTEUTUS

Tässä kappaleessa käydään läpi pelin teon eri vaiheet aina suunnittelemisesta pelintestaukseen.

3.1. projekti

Pelin kehittäminen alkoi osana multimediallmaisurssia syksyllä 2004. Tarkoituksena oli tehdä Flashilla jonkinlainen internetissä toimiva peli. Työryhmän kanssa asiaa pohdittuamme päädyimme lintuperspektiivistä kuvattuun toimintaseikkailuun. Aluksi tarkoitus oli, että peli olisi koostunut ainoastaan yhden ruudun kokoisista piirrettyihin kiinteisiin grafiikoihin perustuviin tehtäviin. Päätimme kuitenkin laajentaa tavoitettamme. Päädyimme useiden ruutujen kokoisin tehtäviin, joissa olisi tavaroita, lukittuja ovia, avaimet niiden avaamiseen, sekä tietysti vihollisia, loppuvastustajaa unohtamatta.

Pelin työryhmään kuuluu Aki Flink (ohjelmointi, musiikki), Vesa Paavilainen (ohjelmointi), Ville Kaipila (grafiikka) ja minä itse (ohjelmointi). Kenelläkään työryhmän jäsenistä ei ollut kovinkaan laajaa aikaisempaa kokemusta Flashista, eikä varsinkaan sen ActionScriptistä.

Kurssin aikana kävi hyvin nopeasti selväksi, että sen puitteissa emme kerkiäisi tätä peliä toteuttamaan. Niinpä päätimme tehdä pelistä pelkän demon multimediallmaisurssin tarpeisiin ja laajentaa pelin ihan omaksi projektikseen, mikä lopulta muodostuikin minun ja Flinkin Akin lopputyöksi.

3.2. Suunnitteluvaihe

Alkuperäisen idean mukaan pelissä piti olla useampia eri tehtäviä. Pelaaja

pystyi etenemään pelissä haluamassaan tehtäväjärjestyksessä. Lopulta tämä idea unohdettiin, sillä useamman tehtävän tekeminen tämän aikataulun puitteissa olisi ollut liian työlästä. Vaikka pelissä olisi ollut useampi hienosti toteutettu tehtävä, suurin osa peliä pelaavista ihmisistä tuskin olisi ikinä nähnyt peliä ensimmäistä tasoa pidemmälle.

Päädyimme lopulta ratkaisuun, missä pelaaja voi valita pelin alussa yhden kolmesta ”maihinnousu” paikasta. Valittavana ovat metsä (forest), ranta (beach) ja aavikko (desert). Kolmen lähtöpaikan ansiosta pelissä on teoriassa kolme erilaista reittiä läpipelaamiseen.

Suunnittelu on erittäin tärkeää. Hyvä suunnitelma on jo puoliksi tehty peli. Suunnitelmat pitää kirjata tarkasti ylös ja niitä tulee noudattaa. Pitää huolehtia siitä, että jokaisella työryhmän jäsenellä on aina ajan tasalla oleva suunnitelma. Toisaalta suunnitelma elää projektin aikana. Aina tulee uusia hyviä ideoita ja innovaatioita, joita kukaan työryhmästä ei edes pystynyt projektin alkupäässä kuvittelemaankaan. Ei siis pidä täysin sokeasti pitää kiinni alussa laaditusta suunnitelmasta. Pitää jättää tilaa myös improvisaatiolle.

On myös tärkeää, että kaikki työryhmän jäsenistä on ajan tasalla mitä muutoksia kuhunkin versioon on tehty. Tätä varten pidimme projektin mukana aina versionhallinta-nimistä tekstitiedostoa, jonka avulla pystyimme selkeästi seuraamaan projektiin tehtyjä muutoksia. (Ks. liite 3.)

3.2.1. Pelihahmon liikuttaminen

Ensimmäisiä asioita, joita peliä varten tehtiin, oli liikkumistoiminto. Aivan ensimmäisissä kokeiluissa käytettiin pelihahmon sijasta ruudulla liikkuvaa neliötä, hieman myöhemmin animoimatonta testihahmoa. Tarvittiin toiminto, joka lukee tietoja näppäimistöltä. Ensimmäiseksi tätä tarkoitusta varten luotiin olio KeyListener. Tämä olio ”kuunteli” näppäimistöä. Kun pelaaja painoi tiettyjä näppäimiä, tässä tapauksessa nuoli ylös, nuoli alas, nuoli oikealle, nuoli vasemmalle, sekä ampumiseen ctrl, tapahtui näppäimen mukainen toiminto. Tämän toteutustavan ongelmana oli se, että kun näppäimiä painettiin yhtä aikaa, ei ohjelma tiennyt mitä tehdä, joten se ei tehnyt mitään. Näin ollen

ampuminen kävelemisen aikana, sekä viistoihin liikkuminen olivat mahdottomia toimintoja. Ongelma korjattiin myöhemmin funktiolla tarkastaNapit, johon palaamme myöhemmin.

3.2.2. Törmäystarkastelu

Törmäystarkastelu seinien kanssa oli yksi hankalimmista ongelmista, joita jouduimme tämän projektin puitteissa miettimään. Aluksi mietimme sen toteuttamista Flashin omalla hitTest() funktiolla, mutta kävi ilmi että se on melko raskas prosessorille. Sen lisäksi olisimme joutuneet luopumaan tilejen käytöstä ja käyttämään kiinteitä piirroksia jokaisessa ruudussa. Tämä olisi ollut erittäin työlästä projektin luonteen huomioon ottaen.

3.2.3. Viholliset

Kiinnostavuuden lisäämiseksi pelissä on kolmen perusvihollisen lisäksi myös ns. pomovihollisia, joiden tappamisesta pelaaja palkitaan suuremmilla pisteillä. Näitä vihollisia ovat sotilaskomentaja ja panssarivaunu. Alkuperäisen suunnitelman mukaan sotilaskomentajan piti olla ystävämielinen hahmo, everstiluutnantti Fägerlund, jonka piti antaa pelaajalle tehtävän kuvaus. Näiden vihollisten lisäksi peliä varten tehtiin myös ninja-hahmo, mutta sen käyttäytyminen olisi poikennut niin paljon muiden käyttäytymisestä, eikä sen koodaamiseen löytynyt aikaa.

3.2.4. Pelimaailma

Lähdin toteuttamaan pelimaailman huoneita niin, että ensiksi laadin ruutupaperille esisuunnitelman kartasta. Sen aloitin kartan toteuttamisen sen vasemmasta yläkulmasta, jossa sijaitsee ranta (beach), yksi pelin aloituspaikoista. Tämän jälkeen aloin tekemään editorilla ruutuja yksi kerrallaan vasemmalta oikealle. Pyrin luomaan pelimaailmaan vaikutelman realistisen tuntuista ympäristöstä, joissa oikeasti olisi metsää, rantaa, autiomaata ja Metacom -rakennuksia.

3.3. Ohjelmointi

Seuraavassa tarkastelemme tarkemmin pelin oleellisimpia toimintoja ja sitä, kuinka ne on toteutettu.

3.3.1. Pelihahmon kontrollointi

Nato Commando pelissä pelaaja ohjaa commando-spriteä pitkin maastoa. Jotta pelihahmo ylipäänsä liikkuu yhtään mihinkään suuntaan, on ohjelman tulkittava näppäimistöä.

Tämä toiminto tehdään funktiolla tarkastaNapit(), jota kutsutaan commando-hahmon enterFrame -eventissä 40 kertaa sekunnissa. Funktio tarkastelee painettuja nappeja, jotka ovat nuoli ylös, nuoli alas, nuoli vasen, nuoli oikea, sekä ctrl. Tämä mahdollistaa pelaajan liikuttelun eri suuntiin, jopa viistoihin. Sen lisäksi ctrl-napin painallus tulkitaan ampumiseksi. Ampuminen on mahdollista myös liikkussa. Rajoitteena on Windowsin oma keyBuffer, joka saattaa pitkän liikkumisen jälkeen tulla täyteen ja liikkuminen pysähtyy. Tämä ongelma ilmenee kuitenkin erittäin harvoin pelin aikana.

Tässä vaiheessa ohjelma vasta tunnistaa napit. Seuraavaksi nappien painelemisen mukaan commando -spritin x- ja y-koordinaatteja muutetaan nopeuden mukaan peliruudulla pelaajan haluamaan suuntaan. Nopeudeksi on lopullisessa versiossa määriteltä 3, mikä tarkoittaa sitä että jokaista napin painallusta kohti hahmo liikkuu kolme pikseliä pelaajan osoittamaan suuntaan.

Jotta liike näyttäisi uskottavalta, on commando -spriteä animoitava. Tähän tarkoitukseen on tehty animateMove() funktio sekä sen käytössä olevat muuttujat. Commando -sprite on yksi movieclip, joka pitää sisällään kaikki animoidut hahmon liikkeet kuhunkin suuntaan. AnimateMoveen lähetään kunkin napinpainalluksen mukana suunta (up, down, left, right). AnimateMove puolestaan käskee commando -spritin aikajanaa siirtymään kunkin napinpainalluksen johdosta yhden ruudun eteenpäin. Näin syntyy illuusio kävelyä muistuttavasta liikkeestä. Tämän lisäksi animateMove tallentaa globaaliin muuttujaan nose tiedon mihin suuntaan hahmo on liikkumassa, mikä

on välttämätöntä jotta luoti osataan suunnata lentämään oikeaan suuntaan.

Kun pelaaja painaa ctrl-nappia, hahmo ampuu. Tämä on toteutettu seuraavasti: Muuttujaan `_global.luoti` liitetään kirjastosta pelimaailmaan `luoti -movieclip` siihen kohtaan missä pelaajan ase on. Tämän jälkeen luodille annetaan aikaisemmin mainitun `nose` muuttujan pohjalta suunta, joka on sama mihin pelaajan nenä osoittaa. Tämän jälkeen asetetaan intervalli, minkä tehtävä on pelaajan aseesta riippuen pitää viive laukausten välissä. Esimerkiksi haulikossa tämä viive on suuri, kun taas konekiväärissä ei käytännössä ole viivettä lainkaan.

3.3.2. Pelimaailman kuvaaminen xml-tiedostoilla

Flashin ohella xml näyttölee tärkeää osaa Nato Commando –pelin toteutuksessa. Käytämme työssämme xml:ää kuvaamaan pelimaailmaa, mikä tarkoittaa kahdenlaisia xml-tiedostoja: `mission.xml` pitää sisällään pelimaailman kartan (Ks. liite 1.). Näitä on vain yksi kappale pelissä. Sen lisäksi pelissä on 113 kappaletta `levelX_Y.xml` -nimisiä xml-tiedostoja, jonka pohjalta luodaan pelikenttä ruutu kerrallaan (Ks. liite 2.). Nimissä X tarkoittaa monennelako rivillä pelikenttä sijaitsee pelimaailmassa ja Y puolestaan tarkoittaa sitä, monesko solu rivillä on kyseessä.

`Mission.xml` koostuu yhdestä pääelementistä `missio`, joka pitää neljä lapsielementtiä:

- `title` kertoo tehtävän otsikon (ei käytössä tämänhetkisessä versiossa pelistä)
- `briefing` kertoo pelin (tai tehtävän) tavoitteet
- `playerstart` pitää sisällään kolme `start-nodea`, joissa määritellään pelaajan aloituspaikka pelimaailmassa pelin alkaessa. Näitä on kolme sen takia, että pelin alussa pelaaja voi valita yhden kolmesta aloituspaikasta pelissä.
- `rooms` pitää sisällään varsinaisen kartan pelimaailmasta. Se sisältää 190 nodia nimeltä `solu`, joilla kontrolloidaan `level.xml`:ien suhdetta pelimaailmassa. Solu pitää sisällään tiedon millä rivillä ja missä solussa

level on pelimaailmassa, sekä levelin ID:n eli toisin sanoen jokaisen levelX_Y.xml tiedostonimen.

LevelX_Y.xml on tarkoitettu peliruudun rakentamiseen. Koska Nato Commando on tile-pohjainen peli, on xml juuri sopiva työkalu peliruutujen kuvaamiseen. Tämä xml koostuu jälleen yhdestä pääelementistä nimeltä pelikenttä. Pelikenttä pitää sisällään kolmea erityyppistä nodea:

- Enemy nodessa määritellään peliruudun vihollisten attribuutit, joita ovat vihollisten määrä, vihollisten määrän satunnaisuus ja vihollisten tyyppi.
- boss nodessa puolestaan kerrotaan ohjelmalle, onko juuri siinä ruudussa vihollispomoa. Jos on, määritellään vihollispomon nimi, eli toisin sanoen tyyppi. Lisäksi määritellään vihollispomon sijainti peliruudulla X ja Y muuttujilla.
- solu node on hiukan saman tapainen kuin sama node mission.xml:ssäkin. Tässä tapauksessa sitä käytetään kuvaamaan rivi ja solu kerrallaan peliruudun koostumus vasemmalta yläreunasta lähtien oikealle alareunaan. Palaan aiheeseen myöhemmin.

Pelimaailman kuvaamisen lisäksi xml:ää käytetään Nato Commandossa pistelistauksen merkintöjen tallentamiseen. Kun peli on loppunut, ladataan palvelimelta Flashiin halloffame.xml -tiedosto, joka pitää sisällään sen 10 sen hetken parasta tulosta, sekä tuloksien tekijöiden nimimerkit. Pelaajan pisteitä verrataan näihin tuloksiin ja mikäli ne ovat tarpeeksi korkeat, pelaaja pääsee kirjoittamaan nimensä pistelistalle. Tämän jälkeen Flashiä komennetaan lähettämään uusi muokattu pistelistaus palvelimella sijaitsevalle PHP-sovellukselle halloffame.php, joka tallentaa päivitetyn version pistelistasta palvelimelle.

(Ks. liite 2.)

3.3.3. Pelimaailman luonti ja liikkuminen ruutujen välillä

Nato Commando -pelin pelimaailma koostuu erilaisista peliruuduista, jotka piirretään ruutuun xml-tiedostojen pohjalta. Kun uusi ruutu halutaan pelaajan

nähtäville, kutsutaan LataaXML() -funktioita. Suoritettaessa LataaXML() lataa Flashin muistiin attribuuttien mukaisen levelX_Y.xml -tiedoston. Tiedoston latauduttua LataaXML() kutsuu funktiota rakennaKentta, missä parsitaan xml:n sisältämä tieto (mm vihollisten määrä ja tyyppi), sekä ajetaan läpi item-map taulukko, johon on tallennettu tieto kaikista poimituista tavaroista (ei siis anneta piirtää uudestaan jo poimittua esinettä).

Tämän jälkeen ajetaan varsinainen piirra() -funktio, joka nimensä mukaan piirtää ruudulle pelikentän. Nato Commandon pelimaailma koostuu tileistä, mikä tarkoittaa tässä tapauksessa 50X50 pikselin kokoisia palikoita, joita piirretään ruudulle vierekkäin vasemmalta yläreunasta oikealle alareunaan. Itse palaset mitä pelissä käytetään löytyvät kirjaston objektista tileSetClip. Paitsi että tämä sisältää kaikkien pelimaailman tilejen grafiikan, se sisältää myös tiedot niiden erilaisista ominaisuuksista, kuten törmäkö pelaaja siihen tai onko esimerkiksi kyseinen tile kerättävissä.

Pelimaailma rakennetaan kolmeen eri kerrokseen. Alimmassa kerroksessa on pohja, toiseen kerrokseen tulevat kaikki mihin pelaaja voi törmätä tai minkä pelaaja voi kerätä, kolmanteen kerrokseen tulevat ne tilet, joiden takaa pelaaja ja viholliset voivat kävellä, esimerkiksi puun latvat. Kolmikerroksisella pelimaailmalla saadaan aikaan kolmiulotteinen vaikutelma peliruudulle.

Koska pelin tehtävässä on useampi eri ruutu, on pelaajan pystyttävä myös liikkumaan niiden välillä. Liikkumisen yhteyteen on tehty tarkastelut, missä katsotaan onko pelihahmo liikkunut ruudusta ulos. Jos näin on, katsotaan mihinkä suuntaan ollaan menossa. Mahdolliset suunnat ovat tietysti ylös, alas, vasemmalle ja oikealle. Jokaista suuntaa kohti on oma funktionsa. Globaaleihin muuttujiin world_rivi ja world_solu on tallennettu tieto pelaajan sijainnista pelimaailmassa. Kun esimerkiksi mennään oikealle, on helppo kertoa mikä xml seuraavaksi ladataan kun world_solu:a kasvatetaan yhdellä. Sama logiikka pätee myös riviin. Kun xml on tunnistettu, se ladataan ja tehdään jälleen uudestaan peliruutu. Mikäli on sattunut jonkinlainen bugi ja pelaaja on päässyt kävelemään ruutuun jota ei ole olemassakaan, pelin suoritus keskeytyy ja viasta ilmoitetaan Flashin output-ikkunassa.

3.3.4. Törmäystarkastelu ja interaktiiviset esineet

Nyt meillä on pelihahmo, joka liikkuu luomassamme pelimaailmassa ruudusta toiseen. Mutta jotta peli olisi kiinnostava, on pelimaailmassa oltava esteitä ja tavaroita joita pelaaja voi kerätä.

Mainitsin aikaisemmin, että pelimaailma koostuu tileistä, jotka ovat määritelty tileSetClipissä. TileSetClipin propertyt-nimisessä layerissä on määritelty jokaisen elementin ominaisuudet. Näitä ominaisuuksia ovat: blocks, collectable, kill, weapon, prize, key, closedDoor ja health.

Kun kenttä luodaan, luodaan samalla taulukko nimeltä instanssitaulukko. Tämä taulukko pitää sisällään jokaisen ruudulla olevan tilen ominaisuudet.

EnterFrame -eventissä tutkitaan jatkuvasti, minne suuntaan pelihahmomme on liikkumassa. Se on helppoa kun tiedämme koko ajan hahmomme jalkojen x- ja y-koordinaatit. Kun nämä koordinaatit jaetaan tilejen leveydellä ja koordinaateilla saamme arvot, joita vertaamalla vastaaviin arvoihin instanssitaulukossa saamme selville elementin ominaisuudet, jonka päälle hahmo on liikkumassa. Mikäli kyseessä on blocks -ominaisuuden omaava tile, hahmoa ei päästetä liikkumaan sen päälle.

Pelin vihollishahmot ja luodit käyttävät samaa tekniikkaa seinään törmäämisen tunnistamiseen.

Samalla tavalla tunnistetaan myös muut elementin muut ominaisuudet. Niiden toteuttamiseen on vain käytetty erillisiä funktioita. Esimerkiksi kun pelaaja saapuu kerättävän esineen luokse, ohjelma tunnistaa että esine on kerättävissä. Esineestä riippuen pelaaja saa lisää pisteitä, hänen aseensa vaihdetaan tai lisätään kerättyihin tavaroihin avain. Tämän jälkeen esine poistetaan peliruudusta, ja itemmap muuttujaan tallennetaan tieto siitä, että tavara on jo kerätty, eikä sitä enää piirretä ruutuun sinne palattaessa.

Lukitut ovet ovat asia erikseen. Itsessään ne eivät ole kerättäviä, eivätkä muullakaan tavalla vaikuta pelaajaan, paitsi pysäyttämällä tämän mikäli hänellä ei ole oikeaa avainta mukanaan. Peli on rajoitettu yhdeksään eri avaimeen. Kun

pelaaja törmää lukittuun oveen, tarkastetaan onko hänellä mukanaan ovikoodin mukaista avainta. Mikäli on, muutetaan kyseisen oven tile aukinaiseksi oveksi ja soitetaan oven mukainen avautumisääni. Pelissä on ainoastaan 9 erilaista avainta koska se oli toteutuksen kannalta helpompaa. Kun pelaaja palaa huoneeseen, jossa avainta on kerran käytetty, ovi on jälleen kiinni, mutta pelaaja voi sitä päin kävelemällä avata sen uudelleen. Jos ovi olisi ensimmäisen avaamisen jälkeen aina auki, olisi joutunut tekemään oman taulukon jokaiselle avatulle ovelle, jotta niiden tila muistettaisiin myöhemmin ruutuun palatessa. Lisäksi on myös realistisempaa että yksi avain käy yhteen oveen ja avain pysyy tallessa, eikä sitä heitetä pois.

3.3.5. Vastustajat

Vastustajat näyttelevät Nato Commandossa tärkeää osaa. Vastustajat generoidaan jokaisessa ruudussa erikseen levelX_Y.xml:stä saatujen tietojen pohjalta. Xml:ssä kerrotaan ohjelmalle vihollisten maksimimäärä, tyyppi ja se, onko vihollisten määrä satunnainen.

Kun vihollisia luodaan ruudulle, huolehditaan aivan ensimmäiseksi siitä, että edellisen ruudun viholliset poistetaan ruudulta. Tämän jälkeen luodaan xml:n mukaisesti viholliset ja samalla tarkastetaan, että ne eivät ilmesty alueille, jotka ovat määritelty blokkaviksi tai liian lähelle pelaajaa.

Varsinaisia vastustajia on ainoastaan kolmea eri tyyppiä: Metacom -mies, terroristi ja Itsemurhapommittaja. Niiden käyttäytyminen määritellään saman skriptin pohjalta erotuksena Itsemurhapommittaja, joka toimii muuten samankaltaisesti, mutta ei ammu ja pelaajan koskiessa hän räjähtää.

Kun vastustaja on asetettu peliruudulle, annetaan sille joukko satunnaisia arvoja, jotka ohjaavat mm. vastustajan käytöstä. Vastustajalle arvotaan muuttuja weapon, mikä määrittelee vastustajan aseensa kahdesta eri vaihtoehdosta (käsiasse ja konekivääri). Samalla määritellään myös muuttujat thisSpeed, joka määrittelee vihollisen nopeuden, sekä aggressivity_shooting ja aggressivity_moving, jotka määrittävät vihollisen aggressiivisuuden ja ampumisen tiheyden.

3.3.6. Miinat

Vihollisten lisäksi pelaaja voi myös kuolla miinaan. Miinat asetetaan peliruudulle muiden elementtien tavoin ja niiden sijainti kerrotaan ohjelmalle xml-tiedoston avulla. Kun pelaaja astuu miinaan, pelaajan vaurio muuttuja nostetaan välittömästi huipulle ja soitetään räjähdysen ääni. Pelaaja kuolee ja peli päättyy. Pelin koodi on myös suunniteltu muita vaurioittavia elementtejä varten, kuten vaikkapa myrkkylätäkkö, tulipalo, piikkilanka-aita.

3.3.7. Luodit

Kun pelaaja painaa tulitusnappia, pelikentälle luodaan luoti, jolle kerrotaan nose-muuttujan avulla mihinkä suuntaan sen tulee lentää.

Kun luoti lentää johonkin suuntaan peliruudulla, jokaisessa ruudussa sitä liikutetaan eteenpäin. Samaan enterFrame eventtiin on rakennettu myös tarkistus, joissa katsotaan, onko luoti osunut seinään tai onko luoti osunut mahdollisesti päin vihollista. Mikäli luoti on osunut seinään, se ainoastaan poistetaan, mutta mikäli luoti osuu viholliseen, tehdään sen yhteydessä muutamia muitakin toimintoja.

Jotta saadaan selville, osuuko luoti peliruudulla olevaan viholliseen, on jatkuvasti suoritettava jokaisen luodin skriptissä hitTest() jokaisen ruudulla olevan vihollisen osalta. Tämä suoritetaan yksinkertaisessa for -loopissa. Mikäli hitTest() palauttaa arvon true jonkin vihollisen kohdalla, lähetetään kyseisen vihollisen id -tunnus funktion die() käsiteltäväksi. Funktio die() toteuttaa vihollisen kuoleman. Aivan ensimmäiseksi määritellään kyseinen vihollinen kuolleeksi, jolloin se ei enää pysty tappamaan pelaajaa, eikä siihen myöskään osu luodit. Seuraavaksi funktio siirtää vihollishahmon suorittamaan kuolinanimaation, mikä alkaa vihollisen aikajanalla ruudussa 75. Tämän jälkeen lisätään pelaajan pisteitä kutsumalla funktiota addScore pääaikajanalla. Sitten vielä soitetään kuolemankorahduksen ääni ja lopuksi poistetaan viholliseen osunut luoti. Vihollisen verinen ruumis jää pelikentälle realismin lisäämiseksi.

Jotta pelissä olisi kunnolla haastetta, on vihollisten pystyttävä vaurioittamaan ja tappamaan myös pelaaja. Vihollisluotien törmäystarkastelu on pelaajan luoteja yksinkertaisempaa, sillä me tiedämme, että pelissä voi olla ainoastaan yksi pelaajan kontrolloima pelihahmo. Commando -movieclipissä tehdään Enterframe -eventissä hitTest() tarkastelu. Mikäli rekisteröidään luodin osuminen pelaajaan, vähennetään pelaajan energiaa. Mikäli pelihahmo osuu vihollishahmoon, toimii vihollishahmon törmäystarkastus, ja pelaajan energiaa vähennetään.

Kun pelaaja saa vauriota, kasvatetaan global.damage muuttujaa ja siirretään sprite_healthbarin aikajanaa yhdellä ruudulla eteenpäin, eli muutetaan energiaa kuvaavaa nato-logoa ruudun oikeassa reunassa punaisemman sävyiseksi. Mikäli osumia on kertynyt 50, käynnistetään pelaajan kuolinanimaatio.

Välipomojen sijainti on määritelty levelX_Y.xml:ssä. Kun ruutuun halutaan välipomo, määritellään xml:ssä pomon tyyppi, sekä sijainti ruudulla. Kun pelaaja saapuu välipomon sisällään pitävään ruutuun, on pelaajalla mahdollisuus ampua tämä ja saada reilu määrä lisäpisteitä.

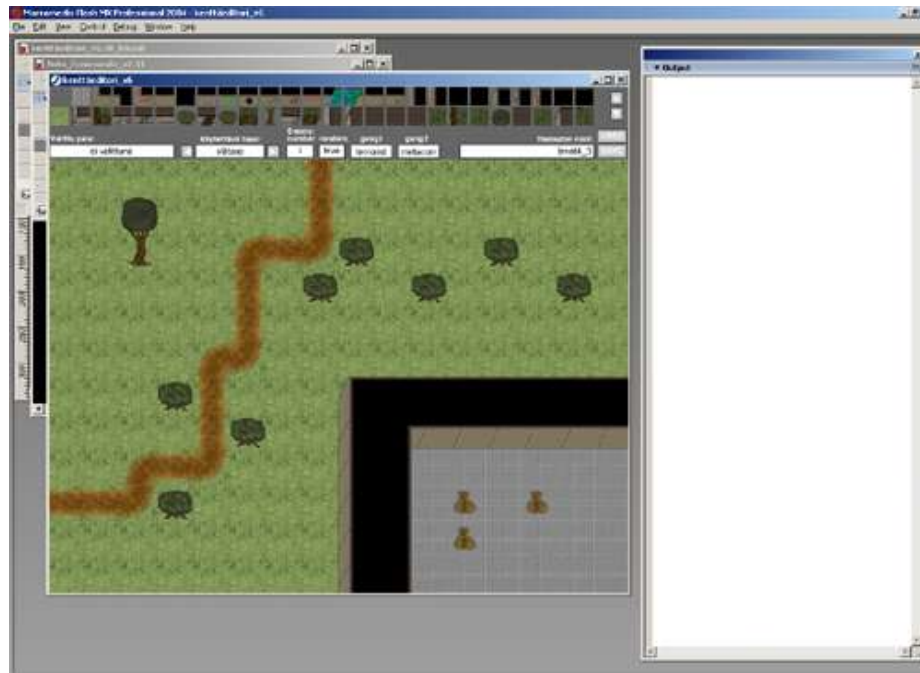
Panssarivaunu on luonnollisesti vielä kestävämpi, kuin muut hahmot. Mikäli pelaaja onnistuu tuhoamaan panssarivaunun, lopetetaan peli ja siirretään pelaaja seuraamaan ilmoitusta pelin läpäisemisestä.

3.4. Pelikenttien rakentaminen

Olemme päässeet nyt siihen vaiheeseen, että meillä on peli, jossa pelaajan ohjaama hahmo liikkuu maastossa, törmäilee esteisiin ja osaa ampua. Sen lisäksi meillä on pelikentällä juoksentelevia vihollisia, jotka ampuvat takaisin ja joita voi päästää päiviltä ampumalla. Mutta siinä on vasta pelin perusrunko. Eräs olennainen asia puuttuu, ja se on pelikenttien suunnittelu.

Toteutin Nato Commandon pelimaailman Aki Flinkin luomalla kenttäeditorilla.

2. pelieditori



Flash -pohjainen editori tarjoaa käyttäjälleen graafisen työkalun peliruutujen suunnitteluun.

Editorin yläpuolella on elementtipaletti, josta käyttäjä voi valita elementin, jonka haluaa ruudulle sijoittaa. Nuolinäppäimiä klikkaamalla saa esiin lisää elementtejä. Alapuolella avautuu itse peliruutu, joka on oletusarvoisesti tyhjä.

Kun käyttäjä on valinnut elementin, jonka haluaa lisätä peliruudulle, on hänen päätettävä mille kerrokselle hän haluaa sen lisätä. Suositeltavaa on ensiksi täyttää editorilla alakerros pohjapalikoilla, joita ovat mm. ruoho, hiekka, tie ja lattia. Pohjakerros on se, joka jää kaiken muun taustalle. Seuraavaksi täytetään keskikerros, johon sijoitetaan mm. ne elementit, joihin pelaaja halutaan törmäävän. Viimeiseksi sijoitetaan yläkerros, johon tulee kaikki ne elementit, jotka peittävät pelikentällä olevat hahmot alleen (esim. puun latvat). Yläkerroksen tarkoituksena on antaa peliruutuun kolmiulotteisuuden tuntua. Kaikkia elementtejä voidaan sijoittaa kaikkiin kerroksiin. Jos käyttäjä esimerkiksi haluaa, että pelihahmot jäävät vesielementin alle, hän voi sijoittaa vettä yläkerrokseen, luoden illuusion veden alaisesta tunnelista tai vastaavasta.

Kun käyttäjä on tyytyväinen luomaansa peliruutuun hän voi vielä määrittellä kyseisen ruudun vihollisten määrän ja tyyppin. Tämän jälkeen käyttäjä painaa

save -nappulaa. Editori generoi rakennetun kenttäkartan pohjalta xml-tiedoston. Editorin versiosta riippuen se joko tallentaa xml:n annetun tiedostonimen pohjalta palvelimelle (online-versio) tai tulostaa generoidun tekstin flashin output -kenttään (lokaaliversio).

Editorissa ei voi myöskään luoda mission.xml-tiedostoa, joten se pitää luoda käsityönä. Se ei ole hankalaa.

Demoversiossa oli yhteensä 13 huonetta, joilla ei pelattavuuden kanssa ollut juuri mitään tekemistä. Huoneet olivat luotu ainoastaan testikäyttöä varten. Lopullisessa versiossa huoneita on yhteensä 113 kpl, joiden suunnitteluun ja toteuttamiseen kului rutkasti aikaa.

Vaikeinta pelimaailman luomisessa oli arvioida bonusesineiden, lisäaseiden ja ensiapulaukkujen määrä ja sijoittelu pelimaailmaan siten, että pelaamisesta tulee mielekästä. Samoja asioita joutui miettimään myös pomo-vihollisten ja avainten kohdalla. Näitä asioita ei voi itse kartan suunnittelussa tarkasti miettiä, peliä pitää testata uudelleen ja uudelleen optimaalisen pelattavuuden saavuttamiseksi.

3.5. Pelitesta

Pelin koodi voi sisältää kymmenittäin bugeja, eli ohjelmointivirheitä. Jatkuva pelin testaaminen koodausvaiheessa on välttämätöntä. Flashiin on rakennettu output -ikkuna, joka antaa käyttäjälle palautetta ohjelman toiminnassa kun sitä testataan. Output-ikkunassa ilmoitetaan mahdollisista virheistä koodissa ja sen lisäksi käyttäjä voi itse määrittellä trace -komennolla ohjelman ilmoittamaan asioita output-ikkunassa haluamissaan kohdissa. Trace -toiminnot näkyvät ainoastaan, kun ohjelmaa ajetaan Flashin test movie -toiminnon kautta.

Mikäli koodissa on virhe, ohjelma ei todennäköisesti toimi lainkaan odotetulla tavalla. Output-ikkunassa ilmoitetaan virhetilanteissa missä ja millä rivillä virhe sijaitsee. Yleensä tämänkaltaiset virheet ovat kielioppivirheitä ja ne ovat helppoja korjata. Paljon hankalampia tapauksia ovat itse ohjelmointivirheet, jotka aiheuttavat ohjelman toimimisen epänormaalisti. Näitä ongelmia on

metsästettävä peliä testaamalla.

Kun ohjelmoija testaa itse peliään, on hänen valmistauduttava tuntikausien tylsään, samoja rutiineja toistavaan rullanssiin. Testivaiheessa tulisi toistaa niitä asioita, jotka eivät toimi tyydyttävästi. Hyvä testi etenee seuraavasti:

- Pelin testaaminen
- Vian paikallistaminen
- Vian korjaaminen
- Aloitetaan alusta

Flash -ympäristössä ohjelmoijalla ei juuri ole muita työkaluja ongelman paikallistamiseen kuin trace -komento. Esimerkiksi kun ohjelmoija yrittää selvittää muuttujien arvoa tietyssä vaiheessa peliä, hän voi kutsua trace-toimintoa tulostamaan muuttujat haluamansa selittävän tekstin kera output-ikkunaan. Tracen avulla on myös helppo selvittää, missä paikassa X- ja Y-koordinaatistossa jokin objekti on pelin tietyllä hetkellä.

Tuntikausien koodaamisen jälkeen ohjelmoija saattaa helposti tulla sokeaksi kirjoittamalleen koodille. Yksinkertaisimmatkin virheet saattavat jäädä väsyneeltä koodaajalta huomioimatta. Koodisokeuden iskiessä on parasta pitää tauko tai/ja kutsua toinen ohjelmoija tutkimaan asiaa mikäli semmoinen on olemassa.

Virheiden korjaamisen lisäksi pelitestauksella on myös toinen tärkeä funktio: Saada tietoa onko peli itsessään pelattava. Tätä asiaa on pelin tekijöiden hyvin vaikea arvioida objektiivisesti, sillä peliä on varmasti tullut pelattua paljon pelin kehittämisen aikana. Tässä vaiheessa onkin hyvä kutsua paikalle kärsivällinen testiryhmä, joka jaksaa pelata peliä useamman kuin kerran ja joka raportoi virheistä kirjoittamalla ne ylös.

4. LOPPUTULOS

Tavoitteena oli rakentaa toimiva peli internetiin. Siinä onnistuttiin hyvin, sillä peli on internetissä ja se toimii. Se, että onko kyseessä hyvä peli, onkin jo toinen juttu. Tässä vaiheessa on hyvä muistuttaa, että kukaan työryhmän jäsenistä, ei projektin alkaessa ollut mikään flash -ekspertti, joten suuri osa peliin käytetyistä viidestä ja puolesta kuukaudesta oli uusien asioiden ja tekniikoiden opettelua.

4.1. Hyvät puolet

Seuraavassa on mainittu asioita pelin toteutuksessa, joihin olen erityisen tyytyväinen.

4.1.1. Pelimoottori

Mielestäni todella hyvää Nato Commando -pelissä on sen joustava pelimoottori, mikä mahdollistaa peliruutujen luonnin, pelaajan liikkumisen niin ruuduissa kuin ruutujen välillä. Näiden toteuttaminen ei todellakaan ollut yksinkertaista tai helppoa. Uskoisin kuitenkin, että saisimme nämä funktiot toteutettua nopeammin ja paremmin, mikäli nyt lähtisimme tekemään niitä uudelleen joltain toista projektia varten.

Hienoa on myös tavaroiden keräily- ja ovien avaimilla avaamis-funktiot ja yleensäkin tapa, millä tunnistetaan, minkä tilen päällä pelaaja kullakin hetkellä on. Ilman tätä tunnistamista, pelin toteutus nykyisellään ei olisi ollut mahdollista

4.1.2. Keinoäly

Hyvää pelissä on myös keinoäly, joskin en anna kehujani täysin varauksetta. Vastustajat käyttäytyvät pääosin loogisesti, joskus jopa älykkäästi. Ne eivät

syöksy aina suoraan tulitukseen, vaan jäävät jopa seinien taakse väijymään. Hienoa on myös se, että vastustajat ”näkevät” pelaajan, kun tämä on samassa linjassa vihollisen kanssa. Toisaalta huonoa tekoälyssä on se, että kaikki viholliset käyttäytyvät saman tekoälyn pohjalta (jokaisella vastustajalla on kuitenkin omat reagoititasonsa ja vauhtinsa). Tekoälyn puute on myös se, että viholliset aktivoituvat tekemään yhtään mitään vasta ”nähtyään” pelaajan. Muuten ne seisovat vain paikallaan. Joka tapauksessa pelin tekoäly on parempi kuin mitä odotin pelin suunnittelun alussa.

4.1.3. Kenttäeditori

Kenttäeditori on hieno lisäys. Vaikka senkin käytettävyydessä olisi paljon parantamista, on se kuitenkin ollut korvaamaton apuväline peliruutujen suunnittelemisessa.

4.1.4. xml

On suuri etu käyttää tämänkaltaisessa pelissä xml-määriteltäviä tile-pohjaisia peliruutuja. Kun systeemi on kerran tehty, käytännössä ei ole mitään rajoja esimerkiksi sille, kuinka iso pelimaailma voi olla tai kuinka monta tehtävää pelissä on. Xml-tiedostot vievät vain n. 10 kilotavua / kpl tilaa palvelimelta ja ne ladataan Flashin muistiin ainoastaan silloin kun niitä tarvitaan. Mahdollisuudet ovat lähes rajoittamattomat.

4.2. Huonot puolet

Nato Commando ei missään tapauksessa ole täydellinen projekti. Puutteita kyllä löytyy.

4.2.1. Grafiikka

Vaikka grafiikka on sinänsä hyvää ja pelin tunnelmaan sopivan retrohenkistä, on siinä huonot puolensa. Ensinnäkin grafiikkaa on liian vähän. Tilejä olisi saanut olla tässä pelissä ainakin kolme kertaa enemmän. Nyt puuttumaan jäivät mm. verkkoaidat, taistelutantereella makaavat ruumiit, kulkuneuvot, ym.

peliruutua elävöittävät yksityiskohdat. Lisäksi esimerkiksi pohjatilen ruohoa ja hiekkaa olisi voinut olla saatavilla erinäköisinä tileinä, jotta maailmaan olisi saanut enemmän vaihtelevuutta. Myös hahmografiikka olisi kaivannut lisää ehostusta, esimerkiksi vihollisilla olisi voinut olla erilaisia kuolinanimaatioita jne. Grafiikan puute on kuitenkin ymmärrettävää, sillä varsinainen graafikko puuttui kokonaan Nato Commando projektista.

4.2.2. Vihollisten määrä

Pelissä on ainoastaan kolme pelikentällä tallustelevaa, vieläpä samankaltaisesti käyttäytyvää vihollista. Nämä eivät varmastikaan riitä kovinkaan kauaa tarjoamaan vaihtelevaa vastustusta nykyajan pelaajalle. Alkuperäisen suunnitelman mukaan vihollisia piti olla enemmänkin. Grafiikat ja suunnitelmat näitä varten olivatkin jo valmiina, mutta aika ei yksinkertaisesti riittänyt niiden käyttäytymisen ohjelmoimiseen.

4.2.3. Optimointi

Ajanpuutteen johdosta koodin optimoiminenkin jäi lähes täysin tekemättä. Optimoinnilla tarkoitetaan sitä, että osia koodista kirjoitetaan uudelleen tehokkaammin. Koodirivejä on vähemmän ja sen suorittaminen on tehokkaampaa. Luonnollisesti optimointia on tapahtunut jatkuvasti pelin tekemisen edetessä, mutta viimeinen silaus jäi valitettavasti tästä tuotannosta puuttumaan.

Ja koska koodia ei ehditty optimoida, saattaa Nato Commandon pelaaminen olla heikommilla tietokoneilla paikoin tuskallisen hidasta. Ruudun päivitys kärsii, mikäli ruudussa on monta vastustajaa ja paljon luoteja ilmassa. Tämä on valitettavaa, mutta edes täydellisellä optimoinnilla ei hidastelua olisi saanut täysin kitkettyä, sillä Flash ohjelmointiympäristönä jättää toivomisen varaa nopeudessa. Flash-lisäosa on loppujen lopuksi melkoinen tehosyöppö.

4.3. projektista

Nato Commando projektina oli ainakin aluksi melko hyvin toimiva toteutus.

Hommat hoidettiin ajallaan ja kunnolla. Kun projekti Flash-kurssin puitteissa loppui ja siitä tehtiin minun ja Aki Flinkin lopputyö, homma hankaloitui, sillä työryhmästä lähti periaatteessa kaksi henkilöä pois. Vesa Paavilainen ja Ville Kaipila siitä huolimatta tekivät kiitettävästi hommansa, mutta helpompaa olisi ollut, mikäli kummatkin heistä olisivat tehneet Nato Commandosta myös lopputyönsä. Nyt käytännössä koko homma kaatui minun ja Akin päälle.

Lisäksi projektissa oli ongelmallista se, että meiltä puuttui kokonaan projektin hallitsija. Vaikka saimme itsekin hyvin järjestettyä omat tapaamisemme ja päätettyä niissä kuka tekee mitäkin, olisi ollut hyvä, mikäli tiimissä olisi ollut joku, joka olisi pistänyt asioihin vauhtia. Olisimme ehkä saaneet pelin toteutettua paremmin ja nopeammin.

4.4. Lisäideoita peliin

Näillä näkymin meillä ei ole aikaa eikä mielenkiintoa alkaa kehittämään peliä eteenpäin, mutta seuraavassa muutamia ideoita, mitä meillä oli tai mitä on tullut esille peliä kehittäessä.

4.4.1. Aikarajoitus

Tämäntyylisissä peleissä on yleensä jonkinlainen aikarajoitus. Tässä pelissä se toimisi hyvin, varsinkin jos nopeasti toiminut pelaaja palkittaisiin bonuspisteillä. Mikäli aika kuluisi loppuun, voisi vaikka pommi räjähtää ja peli loppua. Toisaalta aikarajoituksessa on se huono puoli, että se rajoittaa pelaajan luonnollista halua tutkia peliympäristöä.

4.4.2. Panttivangit ja muut ei-vihamieliset hahmot

Peliä tekisi huomattavasti mielenkiintoisemmaksi, mikäli pelialueella olisi hahmoja, joilla olisi muitakin funktioita, kuin ampua pelaajaa. Panttivankeja vapauttamalla pelaaja voisi saada lisäbonuksia ja samalla panttivanki voisi kiitokseksi kertoa pelaajalla jonkun pelin etenemistä edistävän vihjeen. Pelialueella voisi liikkua vaikka pelaajan liittolaisia, jotka voisivat kertoa pelaajalle vinkkejä etenemiseen. Huikeimmissa visioissa nämä sotilaat voisivat

lähteä pelaajan mukaan ja auttaa häntä matkalla vihollismaan läpi.

4.4.3. Enemmän lisäaseita

Pelissä on tällä hetkellä lisäaseina ainoastaan konekivääri ja haulikko. Lisäaseita voisi olla ainakin sinko, käsikranaatti, liekinheitin ja vaikkapa laserpistooli. Tämä tietäisi reilusti lisää ohjelmointia ja lisäyksiä grafiikkakirjastoon.

4.4.4. Kulkuneuvon käyttäminen

Hieno lisäys peliin voisi olla myös kulkuneuvojen käyttöönotto. Pelaaja voisi ottaa käyttöönsä vaikkapa, moottoripyörän, jeepin, panssarivaunun tai helikopterin. Eri kulkuneuvoilla olisi mahdollisuus päästä eri paikkoihin ja lisäksi ne toimisivat pelaajan suojana. Kun kulkuneuvon osuisi luoti, ei pelaaja vahingoittuisi vaan vain menettäisi kulkuneuvon altaan.

4.4.5. Elämät

Tällä hetkellä pelissä on energia joka kuluu pelaajan saadessa iskuja. Pelaamisen mielekkyyttä voisi parantaa se, että pelissä olisi elämät. Pelaajalla voisi esimerkiksi olla kolme elämää ja lisäksi mahdolliset lisäelämät. Kun pelaajan energiat loppuisivat ja hän kuolisi, peli voisi jatkua muutaman ruudun päästä.

4.4.6. EventTilet

Hieman erikoisempi innovaatio olisi ns. eventTilet. Nämä olisivat käytännössä näkymättömiä tilejä, joihin voisi liittää tiettyjä koodinpätkiä, jotka laukaisisivat erilaisia tapahtumia. Yksi tämänkaltainen tapahtuma voisi olla vaikkapa hälytyksen alkaminen, missä ensin käynnistettäisiin hälytysääniefekti ja sen jälkeen peliruutuun alkaisi vyörymään ulkopuolelta laumoittain vihollisia.

4.4.7. Vihollispesäkkeet

Pelimaailmassa voisi olla vihollispesäkkeitä, esim. teltta, josta sikiäisi jatkuvasti lisää vastustajia peliruudulle. Teltta pitäisi tuhota, jotta vihollisvirta tyrehtyisi. Tämän funktion myötä olisi muokattava vastustajien kuolemakoodia siten, että resurssien vapauttamiseksi vastustajan kuollessa, kirjastosta haettaisi pelikentälle ruumisobjekti kuolleen vihollis-spritin tilalle, mikä ei pitäisi sisällään mitään koodia.

4.4.8. Lisää vaarallisia tilejä

Tällä hetkellä pelissä on yksi ainoa tappava elementti, miina. Sen lisäksi pelissä voisi olla myös muita vastaavia maastoelementtejä. Tuli, happoallas, piikkilanka, laavavirta, kuilu jne. voisivat toimia enemmän tai vähemmän tappavina elementteinä. Porekylpy ja vaikkapa parannus-säde voisivat toimia elämää palauttavina tileinä.

4.4.9. Teleportit ja liikuteltavat sekä tuhoutuvat tilet

Jotta peliin voisi tehdä hienompia puzzle-tyylisiä ongelmia, olisi hyvä olla olemassa tile, joka siirtää pelaajan tiettyyn ruutuun tiettyyn kohtaan. Sen lisäksi tilejen siirtely vaikkapa yhden tilen mitan verran olisi hyvä ominaisuus. Mikäli pelaajalla olisi käsikranaatteja (tai muita pommeja), voisi niitä käyttämällä murskata seiniä. Jälleen yksi ominaisuus, joka tarjoaisi kenttäsuunnitteluun entistä paremmat mahdollisuudet.

4.5. Lopuksi

Takana on valtava työ. Kaikesta huolimatta olen kuitenkin tyytyväinen lopputulokseen. Pelin tekeminen oli kiireestä riippumatta hauskaa ja ennen kaikkea opettavaista. Kiinnostavaa on nähdä, kuinka porukka ottaa pelin vastaan, kun se laitetaan Trash Videon (www.trashvideo.org) nettisivuille pelattavaksi loppukesästä 2005. Kiinnostavaa on myös se, kuinka peli toimii osana samannimisen elokuvan mainoskampanjaa.

Olisiko pelin voinut tehdä paremmin jotain muuta menettely tapaa käyttäen? Todennäköisesti kyllä. Javalla koodatessa olisi päässyt paremmin käsiksi koneen resursseihin, ilman Flashin kaltaista raskasta rajapintaa. Javan käyttö olisi ollut kuitenkin huomattavasti haastavaa ohjelmoinnin kannalta, eikä välttämättä yhtä joustavaa. Toinen hyvä lähestymistapa olisi ollut käyttää ohjelmoinnissa Macromedia Directoria, jolloin lopputulos olisi saattanut olla ainakin tehokkaampi. Director-lisäosa ei kuitenkaan ole yhtä laajalle levinnyt selaimen lisäpalikka kuin Flash tai Java, joten sen käyttö olisi rajoittanut pelaajakuntaa.

LÄHTEET:

Besley, Bhangal, Eden, Ferguson, Monnone, Peters, Rhodes, Young, 2004, Flash Mx 2004 Games Most Wanted, Springer-Verlag

Eseehosting, Support glossary, 2005, saatavilla www-muodossa:
<http://www.eseehosting.com/support/glossary.php/> (luettu 12.4.2005)

iDataExpress, internet glossary, 2005, saatavilla www-muodossa:
<http://www.idataexpress.com/support/glossary.htm> (luettu 14.4.2005)

Imagination, Overview of Smart Graphics, 2004, saatavilla www-muodossa:
<http://www.theopensource.com/osrsvgover.htm> (luettu 28.2.2005)

Makar, Jobe, 2003, Macromedia Flash MX Game Design Demystified, Macromedia Press

Meigs, Tom, 2003, Ultimate Game Design, McGraw-Hill/Osborne

Sham Banghal, 2001, Flash Actionscript, Edita Oyj

Tom Stoper, Game Biz Advice, 2005, saatavilla www-muodossa:
<http://www.sloperama.com/advice.html> (luettu 14.3.2005)

Terena, Guide to Network Resource Tools, 2005, saatavilla www-muodossa:
<http://www.terena.nl/library/gnrt/appendix/glossary.html> (luettu 14.4.2005)

Wikipedia, The Free Encyclopedia, 2004, saatavilla www-muodossa:
http://en.wikipedia.org/wiki/Online_game (luettu 12 .4. 2005)

Wikipedia, The Free Encyclopedia, 2004, saatavilla www-muodossa:
http://en.wikipedia.org/wiki/Macromedia_Flash (luettu 13 .4. 2005)

LIITTEET:

Liite 1:

Tutustu online-pelihin seuraavissa osoitteissa:

<http://www.freeworldgroup.com/games>,

<http://www.freearcade.com/>,

<http://www.heavygames.com/>,

www.electrotank.com

<http://clevermedia.com/game.php>.

Liite 2: mission.xml

```
<?xml version="1.0" ?>
<missio>
  <title>Final Mission</title>
  <briefing>Mission briefing:\nBreak into Metacom compound\nretrieve the
missing DNA-capsules\neliminate enemy leaders</briefing>
  <playerstart>
    <start rivi="2" solu="2" x="370" y="500" />
    <start rivi="1" solu="1" x="370" y="300" />
    <start rivi="3" solu="1" x="70" y="300" />
  </playerstart>
  <rooms>
    <solu rivi="1" solu="1" roomid="level1_1"/>
    <solu rivi="1" solu="2" roomid="level1_2"/>
    <solu rivi="1" solu="3" roomid="level1_3"/>
    <solu rivi="1" solu="4" roomid="level1_4"/>
    <solu rivi="2" solu="1" roomid="level2_1"/>
    <solu rivi="2" solu="2" roomid="level2_2"/>
    <solu rivi="2" solu="3" roomid="level2_3"/>
    <solu rivi="2" solu="4" roomid="level2_4"/>
    <solu rivi="2" solu="5" roomid="level2_5"/>
    <solu rivi="2" solu="6" roomid="level2_6"/>
    <solu rivi="2" solu="7" roomid="level2_7"/>
    <solu rivi="3" solu="1" roomid="level3_1"/>
```

```

<solu rivi="3" solu="2" roomid="level3_2"/>
<solu rivi="3" solu="3" roomid="level3_3"/>
<solu rivi="3" solu="4" roomid="level3_4"/>
<solu rivi="3" solu="5" roomid="level3_5"/>
<solu rivi="3" solu="6" roomid="level3_6"/>
<solu rivi="3" solu="7" roomid="level3_7"/>
</rooms>
</missio>

```

Liite 2: level1_1.xml

```

<?xml version="1.0" ?>
<pelikentta>
<enemy enemies="1" enemyRandom="true" evilGang1="terrorist"
evilGang2="metacom" />
<boss bossName="" bossX="" bossY="" />
<solu rivi="0" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="1" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="2" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="3" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="4" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="5" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="6" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="7" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="8" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="9" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="10" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="11" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="12" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="13" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="14" tile="123" tile2="123" tile3="1" />
<solu rivi="0" solu="15" tile="123" tile2="123" tile3="1" />
<solu rivi="1" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="1" solu="1" tile="123" tile2="123" tile3="1" />
<solu rivi="1" solu="2" tile="123" tile2="1" tile3="1" />

```

<solu rivi="1" solu="3" tile="123" tile2="1" tile3="1" />
 <solu rivi="1" solu="4" tile="123" tile2="104" tile3="1" />
 <solu rivi="1" solu="5" tile="123" tile2="104" tile3="1" />
 <solu rivi="1" solu="6" tile="123" tile2="104" tile3="1" />
 <solu rivi="1" solu="7" tile="123" tile2="104" tile3="1" />
 <solu rivi="1" solu="8" tile="123" tile2="104" tile3="1" />
 <solu rivi="1" solu="9" tile="123" tile2="105" tile3="1" />
 <solu rivi="1" solu="10" tile="102" tile2="1" tile3="1" />
 <solu rivi="1" solu="11" tile="123" tile2="103" tile3="1" />
 <solu rivi="1" solu="12" tile="123" tile2="123" tile3="1" />
 <solu rivi="1" solu="13" tile="123" tile2="123" tile3="1" />
 <solu rivi="1" solu="14" tile="123" tile2="123" tile3="1" />
 <solu rivi="1" solu="15" tile="123" tile2="123" tile3="1" />
 <solu rivi="2" solu="0" tile="123" tile2="123" tile3="1" />
 <solu rivi="2" solu="1" tile="123" tile2="123" tile3="1" />
 <solu rivi="2" solu="2" tile="123" tile2="123" tile3="1" />
 <solu rivi="2" solu="3" tile="123" tile2="105" tile3="1" />
 <solu rivi="2" solu="4" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="5" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="6" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="7" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="8" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="9" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="10" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="11" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="12" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="13" tile="123" tile2="103" tile3="1" />
 <solu rivi="2" solu="14" tile="102" tile2="1" tile3="1" />
 <solu rivi="2" solu="15" tile="123" tile2="106" tile3="1" />
 <solu rivi="3" solu="0" tile="123" tile2="123" tile3="1" />
 <solu rivi="3" solu="1" tile="123" tile2="123" tile3="1" />
 <solu rivi="3" solu="2" tile="123" tile2="107" tile3="1" />
 <solu rivi="3" solu="3" tile="102" tile2="1" tile3="1" />
 <solu rivi="3" solu="4" tile="102" tile2="1" tile3="1" />
 <solu rivi="3" solu="5" tile="102" tile2="1" tile3="1" />

<solu rivi="3" solu="6" tile="102" tile2="1" tile3="1" />
<solu rivi="3" solu="7" tile="102" tile2="1" tile3="1" />
<solu rivi="3" solu="8" tile="102" tile2="1" tile3="1" />
<solu rivi="3" solu="9" tile="102" tile2="1" tile3="1" />
<solu rivi="3" solu="10" tile="102" tile2="1" tile3="1" />
<solu rivi="3" solu="11" tile="102" tile2="1" tile3="1" />
<solu rivi="3" solu="12" tile="102" tile2="1" tile3="1" />
<solu rivi="3" solu="13" tile="102" tile2="1" tile3="1" />
<solu rivi="3" solu="14" tile="102" tile2="1" tile3="1" />
<solu rivi="3" solu="15" tile="123" tile2="103" tile3="1" />
<solu rivi="4" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="4" solu="1" tile="123" tile2="123" tile3="1" />
<solu rivi="4" solu="2" tile="123" tile2="1" tile3="1" />
<solu rivi="4" solu="3" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="4" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="5" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="6" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="7" tile="102" tile2="117" tile3="1" />
<solu rivi="4" solu="8" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="9" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="10" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="11" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="12" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="13" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="14" tile="102" tile2="1" tile3="1" />
<solu rivi="4" solu="15" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="5" solu="1" tile="123" tile2="123" tile3="1" />
<solu rivi="5" solu="2" tile="123" tile2="107" tile3="1" />
<solu rivi="5" solu="3" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="4" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="5" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="6" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="7" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="8" tile="102" tile2="1" tile3="1" />

<solu rivi="5" solu="9" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="10" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="11" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="12" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="13" tile="102" tile2="113" tile3="1" />
<solu rivi="5" solu="14" tile="102" tile2="1" tile3="1" />
<solu rivi="5" solu="15" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="6" solu="1" tile="123" tile2="123" tile3="1" />
<solu rivi="6" solu="2" tile="123" tile2="107" tile3="1" />
<solu rivi="6" solu="3" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="4" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="5" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="6" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="7" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="8" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="9" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="10" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="11" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="12" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="13" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="14" tile="102" tile2="1" tile3="1" />
<solu rivi="6" solu="15" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="7" solu="1" tile="123" tile2="123" tile3="1" />
<solu rivi="7" solu="2" tile="123" tile2="107" tile3="1" />
<solu rivi="7" solu="3" tile="102" tile2="113" tile3="1" />
<solu rivi="7" solu="4" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="5" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="6" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="7" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="8" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="9" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="10" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="11" tile="102" tile2="1" tile3="1" />

<solu rivi="7" solu="12" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="13" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="14" tile="102" tile2="1" tile3="1" />
<solu rivi="7" solu="15" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="8" solu="1" tile="123" tile2="123" tile3="1" />
<solu rivi="8" solu="2" tile="123" tile2="1" tile3="1" />
<solu rivi="8" solu="3" tile="123" tile2="109" tile3="1" />
<solu rivi="8" solu="4" tile="123" tile2="110" tile3="1" />
<solu rivi="8" solu="5" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="6" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="7" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="8" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="9" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="10" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="11" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="12" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="13" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="14" tile="102" tile2="1" tile3="1" />
<solu rivi="8" solu="15" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="9" solu="1" tile="123" tile2="123" tile3="1" />
<solu rivi="9" solu="2" tile="123" tile2="123" tile3="1" />
<solu rivi="9" solu="3" tile="123" tile2="123" tile3="1" />
<solu rivi="9" solu="4" tile="123" tile2="107" tile3="1" />
<solu rivi="9" solu="5" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="6" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="7" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="8" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="9" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="10" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="11" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="12" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="13" tile="102" tile2="1" tile3="1" />
<solu rivi="9" solu="14" tile="102" tile2="1" tile3="1" />

<solu rivi="9" solu="15" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="10" solu="1" tile="123" tile2="113" tile3="1" />
<solu rivi="10" solu="2" tile="123" tile2="123" tile3="1" />
<solu rivi="10" solu="3" tile="123" tile2="123" tile3="1" />
<solu rivi="10" solu="4" tile="123" tile2="107" tile3="1" />
<solu rivi="10" solu="5" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="6" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="7" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="8" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="9" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="10" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="11" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="12" tile="102" tile2="114" tile3="1" />
<solu rivi="10" solu="13" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="14" tile="102" tile2="1" tile3="1" />
<solu rivi="10" solu="15" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="0" tile="123" tile2="123" tile3="1" />
<solu rivi="11" solu="1" tile="123" tile2="123" tile3="1" />
<solu rivi="11" solu="2" tile="123" tile2="123" tile3="1" />
<solu rivi="11" solu="3" tile="123" tile2="123" tile3="1" />
<solu rivi="11" solu="4" tile="123" tile2="1" tile3="1" />
<solu rivi="11" solu="5" tile="123" tile2="110" tile3="1" />
<solu rivi="11" solu="6" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="7" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="8" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="9" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="10" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="11" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="12" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="13" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="14" tile="102" tile2="1" tile3="1" />
<solu rivi="11" solu="15" tile="102" tile2="1" tile3="1" />
</pelikentta>

Liite 3: Versionhallinta.txt

Nato_Commando_v2.55 (Päivittäjä: Aki)
13.04.05

- Törmäystarkastus bullet-spritessä korjattu (If:n sisällä enemy[0] funktioon siirretyn id:n sijaan).
- Bossien huoneissa voi olla nyt muitakin vihollisia ja ne saa hengiltä normaalisti.
- [klo 13:42] Rikoin sitten näköjään eilettain sen miinan äänen --> korjattu.

TEHTÄVIÄ:

- [Ville L] Boss-hahmojen sijoittelu pelimaailmaan
- [Ville L] Kartan suunnittelu ja totutus

BUGIT:

- [AKI] Sain Villen kuvaileman kontrollinmenetyksen aikaiseksi, mutta se tuntui johtuvan enemmänkin koneen väännön loppumisesta kuin ohjelmavirheestä. ELI ampuminen nopeasti naputelle täyttää näppäinpuskurin ja sen jälkeen kaikki tapahtuu melko pitkällä viiveellä --> liikkuminen näyttää todella oudolle.
- [Aki] Hahmot alkavat joskus katoileen kuvasta (sain itse tämän tapahtumaan toistuvasti ts. oli PIRUN hankala kuolla kun vihulaiset katosivat joskus ennenkuin ehtivät ampu ensimmäistäkään ammusta).
- [12.03.2005] En ole nähnyt tätä tapahtuvan hetkeen. Liekö korjaantunut jonkin muutoksen yhteydessä, tai sitten kyse on vain liian vähäisestä tesmailemisesta.
- [13.03.2005] Nyt sain jälleen yhdellä testikerralla tämän onnistumaan... mutta en osaa sanoa miten.
- [11.04.2005] En ole saanut tätä enää tapahtumaan.
- [Vesa] Viholliset kävelevät seinän sisään (joskin enää todella harvoin)
[aki] Näyttävät harppovan läpi seinistä ainoastaan kävellessään takaperin.
Tarkastaako mikään koodi törmäystä hahmo liikkuessa taaksepäin ???
- [Vesa] Viholliset 'tärisivät' jäädessään odottamaan pelaajan hahmoa kulman taakse (tietyt etäisyydet)

- [Vesa] Koko **tun koodi pitäisi kirjoittaa uusiksi ainakin tekoälyn osalta: siinä ei ole mitään logiikkaa! =>kehittelin päässäni vaihtoehtoista ratkaisua, mutta tällä aikataululla en viitsinyt edes yrittää..

 Nato_Commando_v2.54 (Päivittäjä: Aki)
 12.04.05

UUSIA VIHOLLISIA:

=====

* Boss-vihollisia:

Käytetään täyttämällä tiedot kenttä xml:n boss-tagiin:
 (esim. <boss bossName="fagerholm" bossX="650" bossY="450" />)

"tank"

- kestävyys: 50
- nopeus: 2, liikkuu vain sivusuunnassa.
- aseistus: raketit, hitaita, mutta aiheuttavat paljon tuhoa (15 pistettä).

"fagerholm"

- kestävyys: 25
- nopeus: 4
- aseistus: konepistooli

"bomberBoss"

- kestävyys: 15
- nopeus: 4
- ainoastaan pommi, räjähtää lähietäisyydellä. (40 pistettä)

* normaalivihollisia:

Käytetään kuten muitakin normaalivihollisia, nimi jompaan kumpaan kenttä xml:n evilGang-tagisiin: (esim. <enemy enemies="5" enemyRandom="true" evilGang1="bomber" evilGang2="" />)

"bomber"

- kestävyys: kerta laaki kuten muutkin.
- nopeus: 2
- ainoastaan pommi, räjähtää lähietäisyydellä. (20 pistettä)

MUUTA:

=====

- Viiveintervallit tarkastettu ja korjattu itseasiassa kahteen kertaan.

Lähinnä syntaksivirheistä johtuvia toimintavirheitä viivefunktioissa.

- Boss-hahmoille vauriotumis silmukat bullet-spriteen. Sinne piti lisätä erillinen tarkistusmuuttuja (bossInRoom) ilmoittamaan onko ruudussa jokin bossSprite, koska ohjelma karkaili muuten jostain syystä damage funktiossa tankki-ehdon sisälle vaikka objektinimi tuli funktioon oikein. Tämä ilmeni "bomber"-hahmossa siten ettei niitä saanut ampumalla hengiltä millään (varsinainen Jihad-ominaisuus siis :)).
- bullet-spriteen funktiot äänien tuottamiselle. Tankista poistettu älähdykset ennen viimeistä osumaa, tilalla on kolme kimmokeääntä (ei tosin kovin kummoisia). äänifunktiot käyttävät nyt enemyn x-koordinaattia ts. kimmoke ja älähdykset kuuluvat oikeasta paikasta.
- Villen healthkit-koodi lisätty ja miinaan astuttaessa kuolin kiljaisu poistettu.
- Kirjoituskonefontti briefingruudussa (nyt ei Ville K:n tarvitse enää repiä pelihousuja joka testauskerran yhteydessä :P)

TEHTÄVIÄ:

- [Ville L] Boss-hahmojen sijoittelu pelimaailmaan
- [Ville L] Kartan suunnittelu ja totutus

BUGIT:

- [AKI] Sain Villen kuvaileman kontrollinmenetyksen aikaiseksi, mutta se tuntui johtuvan enemmänkin koneen väännön loppumisesta kuin ohjelmavirheestä. ELI ampuminen nopeasti naputelle täyttää näppäinpuskurin ja sen jälkeen kaikki tapahtuu melko pitkällä viiveellä --> liikkuminen näyttää todella oudolle.
- [Aki] Hahmot alkavat joskus katoileen kuvasta (sain itse tämän tapahtumaan toistuvasti ts. oli PIRUN hankala kuolla kun vihulaiset katosivat joskus ennenkuin ehtivät ampua ensimmäistäkään ammusta).
- [12.03.2005] En ole nähnyt tätä tapahtuvan hetkeen. Liekö korjaantunut jonkin muutoksen yhteydessä, tai sitten kyse on vain liian vähäisestä tesmailemisesta.
- [13.03.2005] Nyt sain jälleen yhdellä testikerralla tämän onnistumaan... mutta en osaa sanoa miten.
- [11.04.2005] En ole saanut tätä enää tapahtumaan.

- [Vesa] Viholliset kävelevät seinän sisään (joskin enää todella harvoin)

[aki] Näyttävät harppovan läpi seinistä ainoastaan kävellessään takaperin.

Tarkastaako mikään koodi törmäystä hahmo liikkuessa taaksepäin ?!?

- [Vesa] Viholliset 'tärisyvät' jäädessään odottamaan pelaajan hahmoa kulman taakse (tiedyt etäisyydet)

- [Vesa] Koko **tun koodi pitäisi kirjoittaa uusiksi ainakin tekoälyn osalta: siinä ei ole mitään logiikkaa! =>kehittelin päässäni vaihtoehtoista ratkaisua, mutta tällä aikataululla en viitsinyt edes yrittää..