

Vanhojen konsoleiden ja ROM-pelien muokkausmahdollisuudet

TURUN YLIOPISTO
Tietotekniikan laitos
TkK-tutkielma
Tietotekniikka
Kesäkuu 2026
Joonatan Taurio

TURUN YLIOPISTO
Tietotekniikan laitos

JOONATAN TAURIO: Vanhojen konsolien ja ROM-pelien muokkausmahdollisuudet

TkK-tutkielma, 24 s.
Tietotekniikka
Kesäkuu 2026

Vanhat konsolit ovat monikäyttöisempiä laitteita kuin pelkästään pelikoneita, joille ei enää julkaista pelejä. Konsoleita muokkaamalla niille voidaan löytää monenlaisia uusia käyttötarkoituksia ja niillä voidaan suorittaa myös itse kehitettyjä sovelluksia. Tutkielmassa tarkastellaan erilaisia tapoja muokata konsoleita ohjelmiston ja laitteiston tasolla sekä keinoja laajentaa niiden toiminnallisuutta. Tutkielmassa haetaan myös keinoja muokata konsoleille suunniteltuja ROM-pelejä sekä tutustutaan itse kehitettyihin konsolisovelluksiin.

Tutkielma hyödyntää saatavilla olevaa tieteellistä tietoa ja täydentää sitä modausyhteisöjen tarjoamalla internet-lähteillä, kuten foorumeilla ja blogikirjoituksilla. Tutkimustieto on suurelta osin empiiristä ja peräisin muokkausten kehittäjiltä ja esittelijöiltä sekä osittain tutkielman laatijan omista havainnoista.

Tuloksena esitellään useiden tunnettujen pelikonsolien muokkauksia ja niiden mahdollisuuksia sekä työkaluja ROM-tiedostojen datan muokkaukseen. Tutkielma perehtyy konsolien laitteisto- ja ohjelmistomuokkauksiin, flash-kortteihin, ROM-tiedostojen tallentamiseen sekä ROM-tiedostojen muokkaukseen ulkoisten työkalujen avulla.

Asiasanat: rom, hack, hakkerointi, konsoli, mod, modi, modaus, modifikaatio, muokaus

Sisällys

1	Johdanto	1
2	Taustaa	4
3	Laitteiden modifointi	7
3.1	Aitoustarkistus ja sen ohittaminen	7
3.2	Ohjelmistomuokkaukset	9
3.3	Laitteistomuokkaukset	10
3.4	Flash-kortit	11
3.5	ROM-tiedoston tallentaminen	12
4	ROM-tiedostojen muokkaus	14
4.1	Raa’an datan tarkastelu	14
4.2	Takaisinmallinnus	15
4.3	Työkalut	17
4.4	Homebrew	19
5	Pohdintaa	21
6	Yhteenveto	23
	Lähdeluettelo	25

Kuvat

3.1 DS:n flash-kortteille saatavilla olevia valikkosovelluksia sekä esimerkki omasta sovelluksesta	12
4.1 The Legend of Zelda - Ocarina of Time (N64) -pelin dialogitekstiä HxD-heksadesimaalieditorissa	15
4.2 New Super Mario Bros. -peli tarkasteltuna NDS Editorissa	16
4.3 New Super Mario Bros. -pelin ROM-tiedostosta saatu sound_data.sdat-tiedosto tarkasteltuna VGMTrans-ohjelmassa	18
4.4 Muokattu versio Super Mario 64 -pelistä SM64 ROM Manager -ohjelmassa	19

Taulukot

2.1	Konsolisukupolvet	4
2.2	ROM-hackeja	6

1 Johdanto

Joskus pelatessa vanhoja pelejä pelaaja saattaa tulla miettineeksi, miten peliä voisi parantaa tai mitä siihen voisi lisätä. Moniin peleihin on olemassa joko virallisia tai epävirallisia tapoja muokata niitä. Jotkin pelit tarjoavat virallisen reitin pelin muokkaamiseen eli modaamiseen, kun pelinkehittäjät näkevät siinä taloudellisen hyödyn [1]. Usein kuitenkin modaus on pelin yhteisön ylläpitämää harrastusta ja siihen liittyy pelin tai pelimoottorin tarkempaa tutkimista.

Vanhoissa peleissä ja erityisesti konsolipeleissä muokkaaminen on usein vaikeaa ja poikkeaa tyypillisestä modaamisesta, sillä niiden alustoja ei ole alun perin suunniteltu muokattaviksi. Pelit jaetaan usein kasetteina tai optisina levyinä, joiden sisällä on ROM-tiedosto. ROM-tiedoston muokkauksesta käytetään nimitystä ROM Hacking tai ROM-hakkerointi.

ROM-hakkerointi on aihe, joka on hieman haastava tutkittava, sillä siitä ei löydy kovin paljoa tieteellistä tutkimustietoa varsinkaan suomeksi, ja itse asiassa sille ei ole edes kunnollista suomenkielistä käännöstä. Tässä tutkielmassa tullaan käyttämään termejä "modaus", "ROM-hakkerointi" ja "ROM-tiedoston muokkaus". Valmiista tuotteista eli hakkeroiduista ROM-tiedostoista tullaan käyttämään termiä "ROM-hack" tai "muokattu ROM-tiedosto".

Tutkielman tavoitteena on yhdistää saatavilla olevaa vähäistä tieteellisiä kirjallisuutta ja runsaampaa internetistä löytyvää empiiristä tietoa ja tuoda aihetta paremmin saataville myös tieteellisissä piireissä.

Tässä tutkielmassa haetaan vastauksia siihen, miten pelejä ja pelikokemusta voidaan muuttaa edellä mainittujen muokkausten avulla, ja syvennyttään konsolien muokkauksen teknisiin yksityiskohtiin. Tutkielmassa haetaan myös esimerkkejä, joissa pelejä on muokattu hyvin erilaisiksi alkuperäisestä, sekä esimerkkejä, joissa pelejä on pyritty parantamaan mahdollisimman paljon säilyttäen uskollisuuden alkuperäiselle tekeleelle. Tutkielmassa pyritään myös tutkimaan vanhojen laitteiden monikäyttöisyyttä ja sitä, miten niille voidaan soveltaa mahdollisimman erilaisia käyttötarkoituksia modernien keinojen avulla.

Tutkielmassa perehdytään ensisijaisesti Nintendon konsolien, sekä muutaman valikoidun pelin hakkerointiin, mutta osa käsiteltävistä aiheista on yleispätevää monien muiden laitteiden ja pelien hakkerointiin. Tämän työn tutkimuskysymykset ovat:

TK1 Miten konsolien toiminnallisuutta voidaan muokata tai laajentaa?

TK2 Miten ROM-pelejä voidaan parantaa tai muokata alkuperäisestä?

TK3 Miten ROM-peleihin saadaan lisättyä omaa sisältöä?

TK4 Mitä kaikkea sellaista vanhoilla laitteilla voidaan tehdä, joihin niitä ei ole alun perin suunniteltu?

Tutkielma pohjustetaan tieteellisillä viitteillä ja käsittelyä laajennetaan arkipäiväisemmillä lähteillä. Tutkielmassa hyödynnetään yksittäisiin aiheisiin (kuten järjestelmäpiirien tutkiminen, ROM-dumppaus) niistä saatavilla olevia tieteellisiä lähteitä ja käytetään niitä omissa osioissaan. Merkittävimpinä tietolähteinä modaukseen ja hakkerointiin keskittyvissä osioissa käytetään enimmäkseen käyttäjien omista blogeista sekä internet-foorumeilta löytyviä havaintoja, jotta yksittäisiä modeja voidaan tutkia syvällisemmin ja siinä muodossa, jossa niiden löytäjät ja kehittäjät ovat tuoneet ne julki.

Tietoa etsittiin ACM Digital Librarysta, IEEE Xploresta sekä Science Directistä. Hakuja tehtiin yksittäisistä käsiteltävistä aiheista lauseilla kuten *"/Konsolin*

nimi] + hacking", "Console + Forensic analysis" ja "ROM + dumping". Tietoa etsittiin enimmäkseen kuitenkin suoraan blogipalstojen, foorumien ja medialähteiden hakupalveluita käyttäen sekä yleisillä hakukoneilla. Tietoa haettiin yksittäisten konsolien yksittäisistä modeista käyttämällä lauseita kuten "Nintendo 64 region free mod" ja "DS rom dumping".

Tutkimus jaetaan kahteen pääosioon: konsolien muokkaus ja ROM-tiedostojen muokkaus. Laitteiden modifointi -osiossa (luku 3) etsitään eri konsoleille tunnettuja modeja sekä tapoja laajentaa konsolien toiminnallisuutta. Tutkimuksessa käsitellään vain useimpia Nintendon konsoleita sekä joitain valittuja muita konsoleita. Ensisijaisesti modausta käydään läpi Nintendo DS- ja 3DS-konsolien kautta, mutta myös muita konsoleita käsitellään siinä määrin, kun käsiteltävät aiheet ovat niihin sopivia. Lopuksi käydään läpi sitä, miten modatulla laitteella voidaan saada haltuun ROM-tiedostoja, josta siirrytään niiden muokkaukseen.

ROM-tiedostojen muokkaus -osiossa (luku 4) käsitellään erilaisia keinoja tutkia tiedoston dataa sekä toistuvia kuvioita. Osiossa käsitellään myös sitä, millaisia sovelluksia ROM-tiedostojen ja pelien muokkaukseen on saatavilla ja miten laite- sekä pelikohtaisia työkaluja voidaan hyödyntää muokkauksessa.

2 Taustaa

Pelikonsolit ovat kehittyneet vuosien varrella merkittävästi, ja ne voidaan teknisten ominaisuuksien perusteella jakaa sukupolviin. Jokaisen uuden konsolisukupolven myötä tekniikat ovat kehittyneet ja pelinkehityksen mahdollisuudet laajentuneet. Lähtökohtaisesti konsolit ovat vain pelaamiseen räätälöityjä tietokoneita, mutta etenkin vanhojen konsolien laitteisto eroaa usein merkittävästi saman ajan tietokoneista, mikä tekee niistä uniikin laitteiston osan ja kiinnostavan tutkittavan. Tutkielman kannalta oleellimmat konsolisukupolvet ja niiden merkittävimmät konsolit listataan taulukossa 2.1.

Taulukko 2.1: Konsolisukupolvet

Konsolisukupolvi	Laitteet	Merkittävimmät tekniikat
3. Sukupolvi	Nintendo Entertainment System (NES), Sega Master System	8 bittiä, spritet, taustat
4. Sukupolvi	Super Nintendo (SNES), Sega Mega Drive / Sega Genesis	16 bittiä, skaalaus, rotaatio
5. Sukupolvi	Playstation, Nintendo 64, Sega Saturn	3D-grafiikka, reaaliaikainen video ja ääni
6. Sukupolvi	Playstation 2, Nintendo GameCube, Xbox, Sega Dreamcast	Shaderit, heijastus, "realistinen" grafiikka

Jokaisessa konsolissa on jotain yksilöllistä, jonka myötä konsolin ympärille muodostuu yhteisö, joka tutkii sen tekniikkaa ja yksityiskohtia. Myös konsolisukupolvet saavat yhteisön kiinnostumaan jonkin tietyn ajan tekniikasta sekä siitä, millaisia erilaisia ratkaisuja eri konsoleihin saman sukupolven sisällä on valikoitunut. Modausyhteisöissä pyritään tutkimaan sitä, millaisia teknisiä keinoja peleissä on käytetty pelielementtien saavuttamiseksi ja miten konsolin laitteiston ominaisuuksia on

hyödynnetty. Konsolien laitteistoa pyritään myös soveltamaan omiin käyttötarkoituksiin, joita tässä tutkielmassa tarkastellaan.

Myöhemmissä konsolisukupolvissa pelit ovat siirtyneet enimmäkseen digitaaliseen muotoon, eikä fyysisiä tietovälineitä enää juuri käytetä. Myöhempien sukupolvien laitteet ovat myös teknisiltä yksityiskohdiltaan verrattain samanlaisia keskenään ja eroavat lähinnä ohjelmistojen ja palveluiden osalta. Uusimmille konsoleille, kuten Nintendo Switch 2, ei myöskään ole saatavilla merkittäviä modeja tämän tutkielman kirjoitushetkellä. Sen vuoksi tutkielmassa käsitellään ensisijaisesti vanhoja konsoleita.

Tyypillisesti vanhoissa pelikonsoleissa ja laitteissa pelaaminen aloitetaan syöttämällä laitteen sisään kasetti, levy tai muu tietoväline. Kyseisen tietovälineen sisällä on ROM-tiedosto, joka sisältää pelin koodin, grafiikan ja muun sisällön. Tässä tutkielmassa käsitellään ROM-tiedoston rakennetta ja muokkausta, konsolien laajennettavuutta ja muokattavuutta sekä vaihtoehtoisia tapoja vanhojen laitteiden käyttämiseen ja pelien pelaamiseen.

ROM-hakkeroinnilla tarkoitetaan ROM-tiedoston muokkausta ulkoisilla työkaluilla. ROM-tiedostoja ei ole lähtökohtaisesti suunniteltu muokattavaksi, vaan niiden sisällön on tarkoitus pysyä muuttumattomana pitkään (vrt. uudelleenkirjoitettava muisti), tai jopa koko elinkaarensa ajan. [2]

ROM-tiedostot kuitenkin voidaan dumpata eli kaivaa ulos tietovälineeltä erilaisien työkalujen, modipiirien ja ulkoisten laitteiden avulla [3], jonka jälkeen niitä voidaan muokata vapaasti. Muokkauksessa rajana on ainoastaan tekijän luovuus sekä konsolin tekniset rajoitukset.

Nykyään on saatavilla paljon erilaisia vaikuttavia ROM-hackeja, jotka hyödyntävät erilaisia tekniikoita luovalla tavalla sekä lisää monenlaista sisältöä. ROM-hackit voivat tehdä yksittäisiä pieniä parannuksia peleihin, tai ne voivat lisätä paljon omaa

sisältöä ja muokata pelistä täysin erilaisen alkuperäiseen verrattuna. Maininnan arvoisia ROM-hackeja, joissa näyttäytyy erilaiset tekniikat esitellään taulukossa 2.2.

Taulukko 2.2: ROM-hackeja

Teos	Kehittäjä	Peli	Alusta	Tietoa
Super Mario Odyssey 64	Kaze Emanuar	Super Mario 64	Nintendo 64	Rekreatio Super Mario Odyssey -pelistä Super Mario 64 -pelin sisällä, pelin tasot ja pelimekaniikat portattu toiseen peliin
Newer Super Mario Bros. Wii	Skawo, Newer Team	New Super Mario Bros. Wii	Nintendo Wii	Yhteisön luoma jatko-osa pelille. Täysin uudet tasot ja paljon omaa sisältöä.

Super Mario Odyssey 64 on Nintendo 64 -konsolille graafisesti ja pelimekaanisesti vaikuttava teos. Sen kehittäjä *Kaze Emanuar* on tutkinut Nintendo 64 -konsolin tekniikkaa pitkään ja pyrkinyt kehittämään sille mahdollisimman optimoituja pelejä. Emanuar on esitellyt erilaisia teknisiä saavutuksiaan muun muassa YouTube-kanavallaan¹. *Super Mario 64* -peliin löytyy myös lukuisia yhteisön jäsenten tekemiä hackeja², joissa hyödynnetään Emanuarin ja muiden Nintendo 64 -konsolin tutkijoiden löydöksiä.

Newer Super Mario Bros. Wii on yhteisön luoma jatko-osa *New Super Mario Bros. Wii* -pelille, "uudempi" Super Mario Bros., joka sisältää täysin uudet tasot, uuden maailmakartan ja useita uusia pelimekaniikkoja. Sen pääkehittäjä Skawo on esitellyt monenlaisia löydöksiään vanhoista peleistä ja konsoleista YouTube-kanavallaan³ sekä tuonut julki niiden erikoisia teknisiä ratkaisuja. Newer Team on tehnyt myös *New Super Mario Bros. DS* -pelille Newer-jatko-osan.

¹Kaze Emanuarin YouTube-kanava: <https://www.youtube.com/@KazeN64>

²Super Mario 64 ROM-hackeja: <https://romhacking.com/hacks>

³Skawon YouTube-kanava: <https://www.youtube.com/@Skawo>

3 Laitteiden modifointi

Jotta muokattuja ROM-pelejä, joista esimerkkejä nähtiin luvussa 2, voidaan suorittaa konsolilla, tarvitsee myös itse konsoli usein modata. Tässä luvussa perehdytään siihen, millaisia muokkauksia eli *modeja* eri konsoleille on saatavilla ja mitä niiden asentaminen vaatii.

Riippuen laitteesta siihen voidaan asentaa joko **ohjelmistomodi** tai **laitteistomodi**. Useimpiin Nintendon konsoleihin on saatavilla ohjelmistomodeja, kun taas esimerkiksi PlayStation 2 -konsoli vaatii laitteistomodin [4]. Tyypillisimpiä modeja ovat:

- aluerajauksen ohittaminen (engl. region free mod)
- pelien aitoustarkistuksen ohittaminen
- muokattu laiteohjelmisto (engl. custom firmware, CFW)
- käynnistinojelma (engl. bootloader)
- ohjelmistopäivityksen tuomien rajoitusten kumoaminen

Monissa tapauksissa ROM-pelejä ja muokattua ohjelmistoa pystyy suorittamaan myös modaamattomalla laitteella. Silloin yleensä tarvitaan flash-kortti (engl. flashcart), joka mahdollistaa epävirallisten sovellusten suorittamisen laitteella. Flash-kortteja käsitellään luvussa 3.4

3.1 Aitoustarkistus ja sen ohittaminen

Useille laitteille on olemassa paljon kloonipelejä, joilla ei ole valmistajan sertifikaattia, mutta jotka toimivat laitteella siitä huolimatta. Esimerkiksi aidot Ninten-

don pelit tunnistaa *Nintendo Seal of Quality* -merkinnästä pelikaseteissa. Kloonipelit hyödyntävät erilaisia keinoja laitteen aitoustarkastuksen ohittamiseksi. Kloonit ovat usein tunnistettavissa alkuperäisestä leiman puuttumisen lisäksi myös niiden fyysisen ulkoasun perusteella [5].

Aitoustarkistus toimii myös pelien laadunvalvontana, sillä konsolivalmistajat voivat valita tarkkaan luotetut pelinkehittäjätiimit. Pelivalikoiman suodattaminen ja pelinkehittäjien tarkka valikointi on ollut tärkeää Nintendon ja sen konsoleiden menestykselle. Jos saatavilla on liikaa huonolaatuisia pelejä, yrityksen maine kärsii ja suuri määrä pelejä saattaa jäädä myymättä, kuten Atarille kävi vuonna 1983. Yhdistettynä muihin taloudellisiin vaikeuksiin suuri huonolaatuisten pelien määrä aiheutti videopelialalle laman, joka toi lopun senaikaiselle konsolisukupolvelle. [6] Videopelinkehittäjät ovat myös maininneet laadunvalvonnan olevan yksi parhaista innovaatioista koko videopelialalla [7].

Jos laitteella halutaan suorittaa epävirallisia sovelluksia tai muokattuja ROM-pelejä, täytyy aitoustarkistus pystyä ohittamaan. Konsolivalmistajat tekevät parhaansa yrittäessään estää ohittamisen, ja uudemissa konsoleissa menetelmät ovat monimutkaisempia kuin vanhemmissa, mutta usein viimeistään muutaman vuoden päästä konsolin julkaisusta modausyhteisö keksii siihen tavan.

Eri konsoleissa ohittaminen tapahtuu hyvin erilaisilla keinoilla. Vanhemmissa laitteissa on voinut riittää aitoustarkistuksen suorittavan piirin oikosulkeminen, jonka myötä laitteella voi suorittaa mitä tahansa sille ohjelmoituja pelejä ja sovelluksia. Tämän myötä on siirrytty ratkaisuihin, joissa sekä laitteissa että pelikaseteissa on piirit, jotka kommunikoivat keskenään jatkuvasti, mikä vaikeuttaa ohittamista huomattavasti. Laitekohtaisia modausmenetelmiä käsitellään tarkemmin luvussa 3.3.

3.2 Ohjelmistomuokkaukset

Useisiin konsoleihin on olemassa ohjelmistomodeja. Nintendon konsoleiden modaukseen on saatavilla kattavia ohjesivuja verkko-osoitteissa [*konsolin nimi*].*hacks.guide*. Nämä sivustot ovat osittain samojen yhteisön jäsenten ylläpitämiä ja noudattavat keskenään hyvin samankaltaista formaattia.

Nintendo DSi- ja 3DS-käsikonsoleissa on laajennettuja ominaisuuksia verrattuna alkuperäiseen DS:ään. Niistä tärkeimpiä ovat sovellusten asentaminen laitteen sisäiseen tallennustilaan sekä SD-korttipaikka, jotka tarjoavat uusia mahdollisuuksia modeille. Kumpaankin konsoliin on olemassa CFW-laiteohjelmisto.

DSi-konsolissa on useita haavoittuvuuksia, joista tuorein ja tunnetuin on *Memory Pit*, jonka esitteli GBATemp-foorumin nimimerkki shutterbug2000 vuonna 2019. [8] Haavoittuvuus hyödyntää laitteen mukana tulevan kamerasovelluksen tapaa lukea SD-kortin sisältöä käynnistettäessä ja mahdollistaa oman koodipätkän suorittamisen kortilta. Sen myötä laitteelle voidaan asentaa CFW-laiteohjelmisto sekä käynnistinohjelma, joka ohittaa laitteen alkuperäisen käynnistyksen ja mahdollistaa omien sovellusten suorittamisen. [9]

3DS-konsolissa on myös useita eri haavoittuvuuksia, ja modausmenetelmät riippuvat laitteen mallista ja järjestelmäversiosta. Vaikka Nintendo on ohjelmistopäivitysten myötä pyrkinyt estämään konsoleiden modauksen ja paikkaamaan haavoittuvuuksia, on jokaiselle mahdolliselle 3DS-mallille ja järjestelmäversiolle kuitenkin nykyään olemassa toimiva modausmenetelmä, sillä modausyhteisö on aina löytänyt uuden haavoittuvuuden. [10]

Nintendo 3DS- ja DSi-konsolit ovat jo pitkään olleet elinkaarensa päässä, eikä niihin tule enää päivityksiä. Näin ollen kummankin laitteen kaikki ohjelmistoversiot ovat modattavissa ja laitteiden sisäinen varoitus modien toimimattomuudesta uudemmalla ohjelmistoversiolla, joka ilmenee päivityksiä asentaessa, ei ole enää ajankohtainen.

Nintendo Wii -konsolissa on myös useita tunnettuja haavoittuvuuksia, jotka mahdollistavat laitteen modaamisen. Modauksessa voidaan hyödyntää *ModMii*-työkalua, jonka avulla voidaan valita kaikki halutut muokkaukset, kuten hyödynnettävä haavoittuvuus, käynnistinohjelma ja Homebrew-valikko. Niiden perusteella työkalu luo ohjeet modaukseen ja lataa tarvittavat tiedostot. [11] Myös **Wii U**¹- ja **Switch**² -konsoleille on olemassa modit, joiden avulla niille voidaan Homebrew-valikko. Homebrew-sovelluksia käsitellään luvussa 4.4.

3.3 Laitteistomuokkaukset

Nintendon NES-konsolit ovat aluelukittuja, eli niillä voi pelata vain samalle maantieteelliselle alueelle suunnattuja pelejä. Tämä tarkoittaa, että esimerkiksi eurooppalaisella NES-konsolilla ei oletuksena voi pelata yhdysvaltalaiselle konsolille valmistettuja pelejä. Aluelukitus on toteutettu niin kutsutulla NES10 Lockout Chip -lukituspiirillä. Yksi tyypillisimpiä modeja NESille on aluelukituksen kiertävä modi, joka asennetaan lamauttamalla NES10-piirin toiminta joko oikosulkemalla se tai katkaisemalla siitä oikea pinni. [12]

Super Nintendo (SNES) on huomattavasti vaikeampi modata, sillä eri alueiden konsoleissa on erilainen lukituspiiri, ja piirin täytyy olla aktiivinen, jotta laite toimii. **Nintendo 64:n** ratkaisu on hyvin samankaltainen kuin SNESin. Jos laitteen haluaa yhteensopivaksi usean alueen peleille, täytyy laitteeseen juottaa usean alueen lukituspiirit oikeanlaisen kaavion mukaisesti, jonka esitteli NFGgames -foorumin nimimerkki Yod@ vuonna 2005. [13]

Playstation 2 -konsoliin on saatavilla useita kymmeniä modipiirejä, joissa on erilaiset ominaisuudet. Modipiiri tarkoittaa, että kyseessä on erillinen mikropiiri, joka pitää juottaa laitteeseen, ja näin ollen myös konsolin laitteistorakenne täytyy

¹<https://wiiu.hacks.guide/>

²<https://switch.hacks.guide/>

jossain määrin tuntea. Modipiirit voivat mahdollistaa muun muassa pelien pelaamisen ROM-tiedostoista, kotitekoisten sovellusten suorittamisen sekä mediatiedostojen toistamisen. [14]

3.4 Flash-kortit

Usein laitteen muokkaamisen sijaan voidaan myös käyttää flash-kortteja. Flash-kortit ovat laitteen pelikasettiliittimeen syötettäviä pelikasetin näköisiä laitteita, jotka sisältävät erillisen tallennustietovälineen (yleensä SD-kortti) ja huijaavat laitetta suorittamaan tietovälineellä olevan sovelluksen reaaliajassa kuin se olisi peli. Flash-kortit ovat välttämättömiä etenkin silloin, kun laitteelle ei ole muuten mahdollista asentaa sovelluksia esimerkiksi sisäisen tallennustilan puuttuessa. Tunnettuihin flash-kortteihin kuuluu³:

- Everdrive N8 (NES), Super Everdrive (SNES), Everdrive 64 (Nintendo 64)
- Gateway, Sky3DS, Stargate (3DS)⁴
- R4, Acekard, iTouch, lukuisat R4-kloonikortit (DS)⁵
- Ez Flash (Game Boy, Game Boy Advance, DS)

Flash-kortit voivat hyödyntää laitteelle spesifiin ominaisuuteen liittyvää haavoituvuutta, kuten alkuperäinen R4 ja Nintendo DS:n automaattinen pelin käynnistys, tai naamioitua tunnetuksi peliksi kuten Action Replay ja myöhemmät R4-kloonikortit.

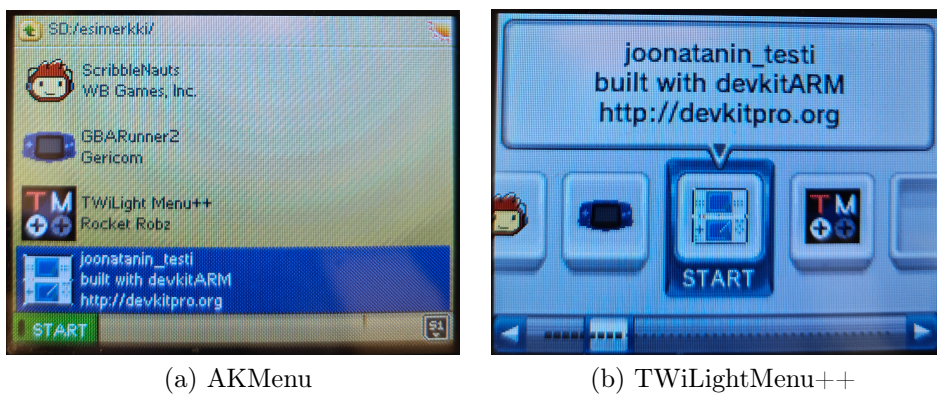
Tyypillisesti flash-korttien sisäänrakennettu laiteohjelmisto sisältää jonkinlaisen valikkosovelluksen, joka toimii tietovälineen tiedostoselaimena. Se mahdollistaa minkä tahansa laitteelle suunnitellun sovelluksen suorittamisen niin, että laite tulkitsee sen kasettipaikkaan syötettynä pelinä, vaikka todellisuudessa kyseessä on tietovä-

³ROM- / Flash-kortteja: <https://retrorgb.com/romcarts.html>

⁴3DS-flashcartteja: https://wiki.gbatemp.net/wiki/3DS_Flashcart_Comparison

⁵DS-flashcartteja: <https://nds.flashcarts.net/card/>

lineellä sijaitseva ROM-tiedosto. Useiden Nintendo DS:n ja 3DS:n flash-korttien tapauksessa käynnistyssovelluksen voi myös mukauttaa. Näille korteille on myös mahdollista asentaa monikäynnistysvalikko (engl. multiboot) sekä useampi eri valikkosovellus, joista vaihtoehtoja on mm. *AKMenu*, *YSMenu* ja *TWiLightmenu++*. Eri valikkosovelluksia, joiden kautta ROM-pelejä sekä omia sovelluksia voi suorittaa, esitellään kuvassa 3.1.



(a) AKMenu

(b) TWiLightMenu++

Kuva 3.1: DS:n flash-korteille saatavilla olevia valikkosovelluksia sekä esimerkki omasta sovelluksesta

Johtuen luontaisesta kyvystä suorittaa mitä tahansa ROM-tiedostoja flash-kortit liitetään yleensä piratismiin [15]. Tämän tutkielman kannalta oleellista on kuitenkin omien sovellusten ja muokattujen ROM-tiedostojen (ROM Hackien) suorittaminen. ROM-tiedostojen hankkimiseen on useita menetelmiä, joista tässä tutkielmassa käsitellään dumpausta.

3.5 ROM-tiedoston tallentaminen

Yksi keino saada haltuun ROM-tiedosto on sen tallentaminen kasetilta tiedostoksi eli dumpaus. Dumpaus on yleinen käsite laitteen tilan tallentamisesta, jota käytetään myös laiteohjelmiston, virhelokien, muistivedosten ja monien muiden sovellusten yhteydessä. Tässä tutkielmassa oleellista on vain pelien dumpaus ROM-tiedostoina sekä laiteohjelmiston dumpaus modien asennuksen kannalta.

Laiteohjelmisto voidaan dumpata ja niin yleensä tehdäänkin modauksen yhteydessä, jotta alkuperäinen laiteohjelmisto saadaan talteen, jos se halutaan joskus palauttaa laitteeseen. Nintendo DSi- ja 3DS-konsolien modausohjeissa kehoitetaan dumppaamaan laitteen laiteohjelmisto järjestelmäpiiriltä varmuuden vuoksi laitetta modatessa. [9], [10]

ROM-tiedostojen dumppaukseen on useita menetelmiä ja ne vaihtelevat konsolin mukaan. Useimmiten apuna tarvitaan erillisiä ulkoisia työkaluja tai flash-kortti. Jos laitteessa on laajennusliitäntöjä, voidaan niitä joissain tapauksissa hyödyntää. Esimerkiksi Game Boyn linkkikaapeliliittimen kautta voidaan dumpata pelikasetin sisältö erilliselle tietovälineelle [16].

Nintendo DS-pelien dumppaukseen voidaan hyödyntää käyttäjän RocketRobz kehittämää *GodMode9i*-työkalua, joka toimii kaikilla DS-malleilla, mutta eri tavalla mallista riippuen. DS ja DS Lite -malleilla, joissa on 2 korttipaikkaa, voidaan slot-1-paikkaan asetetut pelit tallentaa käyttäessä slot-2-paikasta käynnistettyä flash-korttia tai päinvastoin. DSi-konsolilla työkalu voidaan käynnistää Unlaunch-käynnistinohjelman kautta ja siten pelit voidaan tallentaa SD-kortille. 3DS-konsolilla tarvitaan moderni muokattu laiteohjelmisto, jotta pelit voidaan tallentaa. [17]

4 ROM-tiedostojen muokkaus

Tässä luvussa perehdytään muokattujen ROM-tiedostojen eli ROM Hackien, joista esimerkkejä nähtiin luvussa 2, tekniseen puoleen. Kun ROM-tiedosto on saatu laitteelta ulos luvun 3.5 mukaisesti, voidaan sille tehdä muokkauksia ulkopuolisin keinoin. Muokattuja ROM-tiedostoja voidaan suorittaa luvun 3 mukaisesti sopivaksi todetulla laitteella tai emulaattorilla.

4.1 Raa'an datan tarkastelu

Tutkimalla ROM-tiedostoa heksadesimaalieditorissa voi sen datasta löytyä yhteneviä ja toistuvia kuvioita. Niiden avulla voidaan päätellä, minkätyyppistä dataa on kyseessä ja mihin osaan peliä se kuuluu.

Helpoiten löydettävä tietotyyppi on ASCII-koodattu raaka teksti (engl. plain text), sillä useimmat heksadesimaalieditorit näyttävät tekstimuotoisen datan valmiiksi. Joissain tapauksissa jotkin tekstit pelissä kuten valikot, objektien nimet tai dialogit voivat olla tallennettu raakana tekstinä, kuten kuvassa 4.1 nähdään ja siten ne ovat helposti löydettävissä.

Datan seasta voi myös löytyä tunnettuja kuvioita, jotka paljastavat, minkätyyppistä data on, ja kokeneemmat käyttäjät pystyvät siten pääättelemään, miten dataa kannattaa yrittää tulkita. Kokeneet datan tutkijat voivat tunnistaa raa'an datan seasta toistuvista kuvioista esimerkiksi väripalettien arvoja, 3d-grafikan elementtejä tai tunnetun äänimoottorin rakenteita.

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00910000 61 72 74 20 20 20 31 30 20 52 75 70 65 65 73 09 art 10 Rupees.
00910010 01 01 1B 05 42 42 75 79 01 44 6F 6E 27 74 20 62 ....BBuy.Don't b
00910020 75 79 05 40 02 00 00 00 08 49 27 6D 20 73 6F 72 uy.@....I'm sor
00910030 72 79 2E 2E 2E 49 20 63 61 6E 27 74 20 73 65 6C ry...I can't sel
00910040 6C 20 74 68 69 73 20 74 6F 01 79 6F 75 20 75 6E l this to.you un
00910050 6C 65 73 73 20 79 6F 75 20 68 61 76 65 20 61 20 less you have a
00910060 62 6F 74 74 6C 65 20 74 6F 20 01 70 75 74 20 69 bottle to .put i
00910070 74 20 69 6E 2E 09 0A 02 13 20 08 59 6F 75 20 63 t in.....You c
00910080 61 75 67 68 74 20 61 20 05 41 50 6F 65 20 05 40 aught a .APoe .@
00910090 69 6E 20 61 20 62 6F 74 74 6C 65 21 09 01 53 6F in a bottle!..So
009100A0 6D 65 74 68 69 6E 67 20 67 6F 6F 64 20 6D 69 67 mething good mig
009100B0 68 74 20 68 61 70 70 65 6E 21 02 00 13 1A 08 59 ht happen!....Y
009100C0 6F 75 20 67 6F 74 20 05 41 4C 6F 6E 20 4C 6F 6E ou got .Alon Lon
009100D0 20 4D 69 6C 6B 05 40 21 09 01 54 68 69 73 20 6D Milk.@!..This m
009100E0 69 6C 6B 20 69 73 20 76 65 72 79 20 6E 75 74 72 ilk is very nutr
009100F0 69 74 69 6F 75 73 21 01 55 73 65 20 69 74 20 77 itious!.Use it w
00910100 69 74 68 20 05 46 A1 20 05 40 74 6F 20 72 65 63 ith .F; .@to rec
00910110 6F 76 65 72 20 79 6F 75 72 20 01 6C 69 66 65 20 over your .life
00910120 65 6E 65 72 67 79 21 04 13 1A 59 6F 75 20 77 69 energy!...You wi
00910130 6C 6C 20 72 65 63 6F 76 65 72 20 05 41 66 69 76 ll recover .Afiv
00910140 65 20 68 65 61 72 74 73 05 40 01 70 65 72 20 64 e hearts.@.per d
00910150 72 69 6E 6B 2E 20 54 68 65 72 65 20 61 72 65 20 rink. There are
00910160 74 77 6F 01 64 72 69 6E 6B 73 20 70 65 72 20 62 two.drinks per b
00910170 6F 74 74 6C 65 2E 04 13 1A 4F 6E 20 74 68 65 20 ottle....On the
00910180 05 46 53 65 6C 65 63 74 20 49 74 65 6D 20 53 75 .FSelect Item Su
00910190 62 73 63 72 65 65 6E 05 40 2C 01 79 6F 75 20 63 bscreen.@,.you c
009101A0 61 6E 20 73 65 74 20 69 74 20 74 6F 20 05 46 A7 an set it to .FS
009101B0 05 40 2C 20 05 46 A6 20 05 40 6F 72 20 05 46 A8 .@, .F; .@or .F"
009101C0 05 40 2C 01 61 6F 64 20 74 68 65 6F 20 75 73 65 @ and then use

```

Kuva 4.1: The Legend of Zelda - Ocarina of Time (N64) -pelin dialogitekstiä HxD-heksadesimaalieditorissa

4.2 Takaisinmallinnus

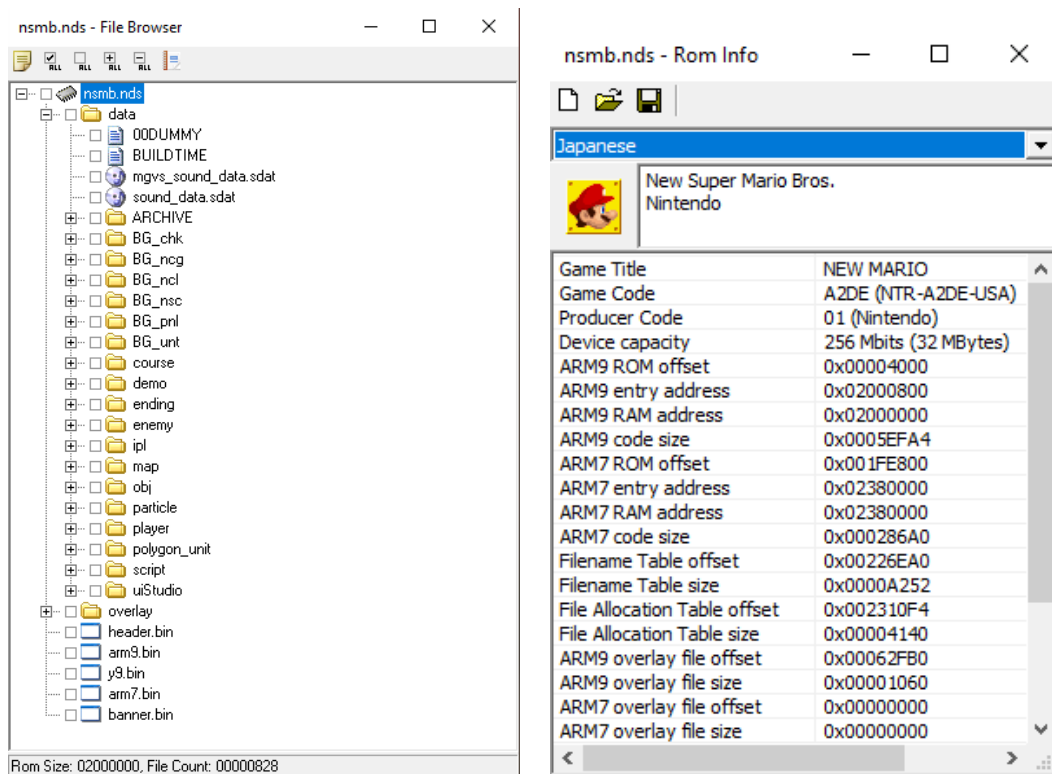
Takaisinmallinnus (engl. reverse engineering) on prosessi, jossa laitteen mikrokoodia tai raakaa dataa tutkimalla yritetään saada muodostettua alkuperäinen ohjelmakoodi tai järjestelmä. Takaisinmallinnus on hyvin laaja käsite ja siksi tässä tutkielmassa tarkastellaan sitä vain ROM-hakkeroinnin näkökulmasta.

Takaisinmallintaessa täytyy tuntea hakkeroitavan laitteen mikroarkkitehtuuri hyvin tarkasti, jotta sille voi muodostaa toimivaa koodia. Esimerkiksi Nintendo 64-konsoli käyttää MIPS-prosessoriarkkitehtuuria ja Reality-apuprosessoria [18], joiden mikrokoodi täytyy tuntea konsolille tehtyjä ROM-pelejä takaisinmallintaessa. Nintendo DS puolestaan käyttää ARM9- sekä ARM7-arkkitehtuureihin perustuvaa

tuplaprosessoriratkaisua, ja kummankin prosessorin mikrokoodia pyritään tulkitsemaan DS:n ROM-tiedostoista.

Vaikka tuntisi laitteen arkkitehtuurin, on jokaisen pelin rakenne kuitenkin aina hieman erilainen ja siksi myös hakeroitavan pelin rakennetta täytyy tutkia erikseen. Samalle laitteelle kehitettyjen pelien välillä on kuitenkin usein joitain yhtenevyyksiä, joten toisten saman laitteen pelien tuntemuksesta on hyötyä.

Tyypillisesti tavoitteena on löytää ROM-tiedostosta rakenteita, jolla pelin sisältöelementit on tallennettu. Uudempien konsolien, kuten Nintendo DS:n (tässä kontekstissa DS, 2004, luokitellaan uudempiin konsoleihin), ROM-tiedostoissa on usein jokin tiedostojärjestelmä (Kuva 4.2), mikä helpottaa eri osien löytämistä ja muokkaamista.



(a) Tiedostojärjestelmä

(b) ROM-tiedoston yksityiskohdat

Kuva 4.2: New Super Mario Bros. -peli tarkasteltuna NDS Editorissa

Vanhemmissa laitteissa, kuten Nintendo 64, taas tiedostojärjestelmää ei yleensä ole, vaan elementtejä löydetään muilla keinoilla, kuten etsimällä pointer-arvoja.

Esimerkki onnistuneesta suuresta takaisinmallinnusoperaatiosta on Super Mario 64 -pelin lähdekoodin käänteismallinnus (engl. decompilation). Projekti mainittiin ensi kertaa GBATemp-foorumilla vuonna 2016 ja valmis lähdekoodi julkaistiin GitHubissa vuonna 2022. Mallinnetun lähdekoodin myötä Super Mario 64 -peli on portattu lukuisille eri laitteille kuten Sony PSP:lle¹, Apple Watch -älykellolle² ja jopa PDF-tiedostossa ajettavaksi koodiksi³.

4.3 Työkalut

Kun ROM-tiedoston rakenteesta on saatu selvitettyä jonkin tietyn datan tyyppi ja sijainti, voidaan sen muokkaukseen kehittää sovellus. Moniin modausyhteisössä tunnettuihin peleihin on olemassa graafisia muokkausohjelmia ja siksi datan muokkaus raakana ei välttämättä ole tarpeellista. Yleisimpiä muokkausohjelmia eli editoreita on erilaiset grafiikka- ja tasoeditorit sekä tekstieditorit. Editorien avulla myös pelin koodia tai ROM-tiedoston rakennetta tuntemattomat käyttäjät voivat muokata peliä ja lisätä siihen omaa sisältöä.

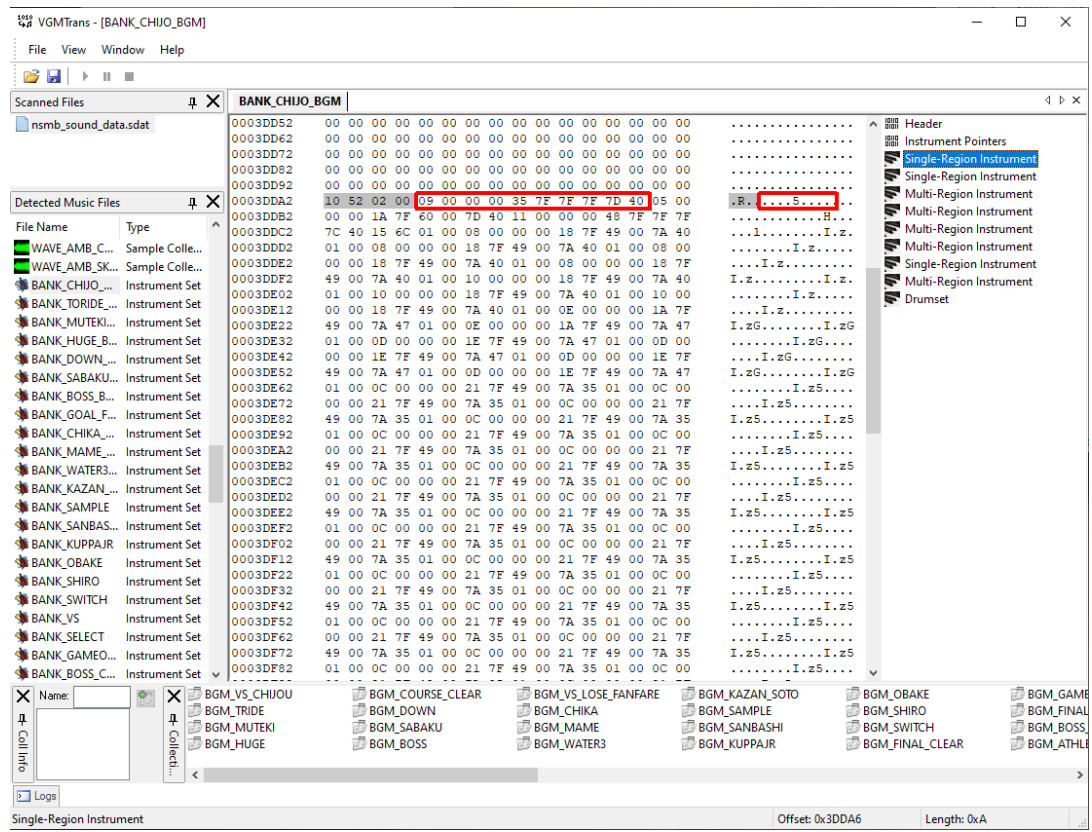
Laitekohtaiset työkalut

Monissa tapauksissa jotkin rakenteet voivat olla toistuvia kaikille tai useille saman konsolin ROM-tiedostolle. Kuvassa 4.2 voidaan havaita *sound_data.sdat*-tiedosto pelin tiedostojärjestelmän juuressa. Se on tyypillinen äänidatan formaatti DS:n ROM-tiedostoissa, ja sen muokkaukseen on olemassa työkaluja. Tiedoston sisältö tarkasteltuna VGMTrans-ohjelmassa voidaan nähdä kuvassa 4.3.

¹Super Mario 64 PSP Port: <https://www.psx-place.com/threads/update-2-super-mario-64-ported-to-vita-psp-ps2-ps3-and-ps4.30923/>

²Super Mario 64 Apple Watch Port: <https://www.techeblog.com/super-mario-64-apple-watch-mod-game/>

³Super Mario 64 PDF Port: https://gigazine.net/gsc_news/en/20260203-super-mario-64-pdf/

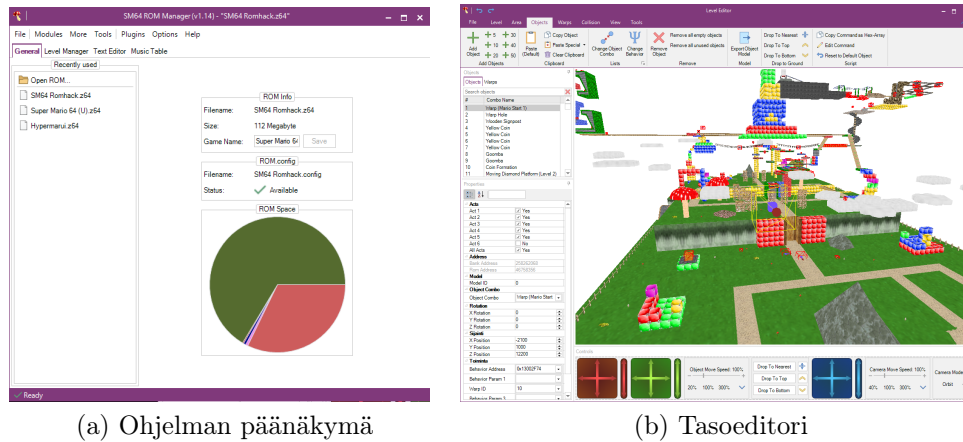


Kuva 4.3: New Super Mario Bros. (Nintendo©) -pelin ROM-tiedostosta saatu sound_data.sdat-tiedosto tarkasteltuna VGMTrans-ohjelmassa

Pelikohtaiset työkalut

Koska ROM-tiedostojen rakenne on lähtökohtaisesti uniikki, on suurin osa muokkausohjelmista suunniteltu vain yhden tietyn pelin muokkaukseen. Jos pelin rakenne on hyvin tunnettu, muokkausohjelma voi sisältää laajatin vaihtoehdot pelin eri osien muokkaamiseen.

Esimerkiksi Super Mario 64-pelin muokkaukseen on olemassa paljon työkaluja, joista yksi merkittävimpiä on SM64 ROM Manager, joka nähdään kuvassa 4.4. Se sisältää useita lisäosia, joiden avulla lähes kaikkea pelin sisältöä voidaan muokata, ja mm. kehittyneen tasoeditorin.



Kuva 4.4: Muokattu versio Super Mario 64 -pelistä SM64 ROM Manager -ohjelmassa

4.4 Homebrew

Perinteisten ja muokattujen ROM-tiedostojen lisäksi laitteilla voidaan suorittaa myös täysin itse kehitettyjä laitteelle yhteensopivia sovelluksia. Homebrew on yleisnimitys kaikille käyttäjien kehittämille omille ”kotipolttoisille” yhteisösovelluksille, joita modatuille laitteille voi asentaa. Kun sovellukset pakataan oikeanlaiseen formaattiin, tyypillisesti muistuttaen laitteen omaa ROM-tiedostoa, ne voidaan suorittaa laitteella.

Joillekin laitteille kuten Nintendo Wiille ja Wii U:lle on olemassa sovellusvaliko⁴ (Homebrew Channel), joka luo helpon tavan käynnistää kaikkia yhteisösovelluksia, sekä yhteisön ylläpitämä sovelluskauppa tai -selain (Wii Homebrew Browser⁵, Wii U Homebrew App Store⁶). Niiden tarkoitus on luoda yhteisösovelluksille keskus, joka muistuttaa toimintaperiaatteeltaan paljolti mitä tahansa sovelluskauppaa. Sovelluksia voi ladata ja käynnistää valikoimasta ja niihin voi ladata päivityksiä.

⁴Kotipolttoisista: https://oscwii.org/library/app/homebrew_browser

⁵Wii Homebrew Browser: <https://hbc.hackmii.com/>

⁶Wii U Homebrew App Store: <https://hb-app.store/>

Erityisesti Nintendon omien sovelluskauppojen sulkeuduttua (2019⁷, 2023⁸) yhteisön ylläpitämät homebrew-sovelluskaupat ovat olleet suosiossa.

Omien Homebrew-sovellusten kehittämiseen on myös työkaluja. Yksi merkittävimmistä kehitysympäristöistä on DevkitPro, joka tarjoaa työkalut muun muassa Nintendo Switchin, Wiin ja DS:n sekä useiden muiden konsolien sovellusten kehittämiseen. Kuvassa 3.1 sivulla 12 nähdään esimerkki DevkitPron avulla luodusta sovelluksesta Nintendo DS:lle.

⁷Wii Shop Channel -sulkemisilmoitus: https://en-americas-support.nintendo.com/app/answers/detail/a_id/27560/~wii-shop-channel-discontinuation

⁸Wii U eShop -sulkemisilmoitus: <https://www.nintendo.com/au/support/articles/wii-u-and-nintendo-3ds-eshop-discontinuation/>

5 Pohdintaa

Tutkielmassa käy ilmi, että konsoleiden ja pelien modaus- ja hakkerointimenetelmiä on paljon. Jopa hyvin rajallisesti tarkasteltuna voidaan löytää useita muokkausmahdollisuuksia tutkielmaan valikoiduille konsoleille ja peleille.

Tutkielma käy läpi pintapuolisesti usean eri konsolin modausta ja sitä, mitä niillä pyritään saavuttamaan. Erilaisia keinoja muuttaa tai laajentaa konsoleiden toimintaa pyrittiin etsimään. Tulokset osoittavat, että kaikkiin tutkielmassa käsiteltyihin konsoleihin on olemassa ainakin yksi modi, joka laajentaa niiden toiminnallisuutta jollain tavalla. Tutkielmassa kävi myös ilmi, että useiden käsiteltyjen laitteiden rajoituksista on mahdollista päästä eroon modausmenetelmillä, vaikka joissakin laitteissa menetelmät ovat vaikeampi toteuttaa kuin toisissa. Tämä on tärkeä havainto, sillä sen myötä voidaan soveltaa ohjelmistokehitystä laajemmin vanhoille konsoleille tilanteissa, joissa rajoitukset olisivat muuten esteenä. Tutkielma mahdollistaa siten jatkotutkimuksen yksittäisten konsoleiden sovelluskehityksestä modit huomioiden.

Kaikki tutkielmassa läpikäytyt modit on käsitelty siten, kuin ne ovat olleet kirjoitushetkellä. Konsoleita, jotka ovat tutkielman mukaan elinkaarensa päässä, voidaan katsoa olevan jokin *de facto* -tapa käyttää modattuna. Esimerkiksi Nintendo DSi- ja 3DS-konsoleihin voidaan asentaa uusimpaan ohjelmistoversioon perustuva laiteohjelmisto helpoimman tunnetun haavoittuvuuden kautta, ja siten avata kaikki tunnettu modauksella laajennettu toiminnallisuus. Tutkielmassa kuitenkin nostetaan esille myös modausyhteisöjen aktiivisuus ja modien jatkuva kehitys, mikä tarkoit-

taa, että konsoleita ei välttämättä koskaan ole parasta tapaa käyttää modattuna, vaan uusia modeja voi aina tulla.

Tutkielmassa ei perehdytä konsoleiden teknisiin yksityiskohtiin tai siihen, miten modaukset ovat teknisesti toteutettu, mutta annetaan pohjustusta jatkotutkimukseen. Tutkielmassa ei myöskään käsitellä modauksen saatavuutta tarkemmin, vaikka modauskeinojen helpottuminen ajan myötä aktiivisen modausyhteisön ansiosta tuodaan esille.

6 Yhteenveto

Useille vanhoille konsoleille on saatavilla modeja, joilla niiden toimintaa voidaan laajentaa. Myös useita pelejä on keinoja muokata ja lisätä niihin omaa sisältöä. Tutkielmassa etsittiin erilaisia mahdollisuuksia muokata vanhoja konsoleita ja pelejä. Jopa yli 30 vuotta vanhoista peleistä tehdään jatkuvasti uusia modattuja versioita. Tässä tutkielmassa käytiin läpi kirjoitushetkellä merkittävimpiä uusia ja vanhoja modeja ja löydettiin vastaukset kysymyksiin:

TK1 Miten konsoleiden toiminnallisuutta voidaan muokata tai laajentaa?

Nintendo NES-, SNES- ja 64-konsoleiden aluerajaus voidaan ohittaa laitteistomodoin avulla ja siten pelivalikoimaa voidaan laajentaa. Nintendo DSi- ja 3DS-konsoleihin voidaan asentaa ohjelmistomodi, joka mahdollistaa omien kotipolttoisten sovellusten ja epävirallisten ROM-pelien suorittamisen.

TK2 Miten ROM-pelejä voidaan parantaa tai muokata alkuperäisestä?

Tutkimalla ROM-tiedoston dataa voidaan joitakin muokkauksia tehdä tiedostoon suoraan. Monille tunnetuille peleille, kuten Super Mario 64 ja New Super Mario Bros., on olemassa lukuisia valmiita muokkaustyökaluja, jotka helpottavat muokkausta merkittävästi.

TK3 Miten ROM-peleihin saadaan lisättyä omaa sisältöä?

Muokkaustyökalut mahdollistavat oman sisällön lisäämisen. Riippuen pelistä työkalujen avulla voidaan lisätä omia tasoja, graafisia elementtejä, 3D-malleja, äänitiedostoja, koodipätkiä sekä paljon muuta. Jotkin työkalut tarjoavat muokkausmahdollisuuden usealle elementille saman pelin sisällä ja jotkin taas tarjoavat muokkausmahdollisuuden yhdyntätyypiselle datalle usean pelin sisällä.

TK4 Mitä kaikkea sellaista vanhoilla laitteilla voidaan tehdä, joihin niitä ei ole alun perin suunniteltu?

Modauksia hyödyntämällä laitteita voidaan käyttää omien sovellusten suorittamiseen, median toistamiseen sekä emulaattoreiden kautta muiden alustojen pelien pelaamiseen. Myös homebrew-sovellusvalikot ja -kaupat mahdollistavat yhteisön kehittämien sovelluksien lataamisen konsoleille. Laitteen tekniikkaa hyödyntämällä siitä voidaan saada paljon laajempi käyttöalusta kuin vain pelikonsoli.

Tutkielmassa käytiin läpi sitä, miten konsoleiden toiminnallisuutta voidaan laajentaa ja havaittiin, että useissa tapauksissa konsoleilla on mahdollista suorittaa omia sovelluksia sekä muokattuja ROM-tiedostoja joko muokatun laiteohjelmiston tai flash-korttien avulla. Tutkielmassa kävi myös ilmi, että tunnetuille peleille on olemassa modausyhteisön kehittämiä sovelluksia, joilla peleihin voidaan lisätä omaa sisältöä ja pelejä voidaan tarvittaessa muokata myös raakaa dataa manipuloimalla.

Tutkielma tarjoaa pohjaa jatkotutkimukselle yksittäisistä konsoleista ja modeista. Tutkimustietoa yhdistettynä konsoleiden tekniikkaan liittyviin dokumentteihin voisi käyttää jonkin tietyn konsolin modauksen tarkemmassa tutkimisessa. Jatkotutkimuksen mahdollisuuksia löytyy myös peliyhteisöistä, jotka modattujen pelien ympärillä toimii sekä modauksen ja modattujen pelien saatavuudesta.

Lähdeluettelo

- [1] O. A. Lev Poretski, "Placing Value on Community Co-creations: A Study of a Video Game 'Modding' Community", *ACM Games*, vol. 4, nro 1, 2017. DOI: 10.1145/2998181.2998301.
- [2] M. J. Flynn, "Read-only memory (ROM)", teoksessa *Encyclopedia of Computer Science*. John Wiley ja Sons Ltd., 2003, s. 1503, ISBN: 0470864125.
- [3] R. Kearey. "A 1970s Mask ROM MCU Spills Its Secrets", viitattu 14. huhtikuuta 2025. url: <https://hackaday.com/2023/09/30/a-1970s-mask-rom-mcu-spills-its-secrets/>.
- [4] J. Grand et al., "Chapter 9 - Hacking the PlayStation 2", teoksessa *Hardware Hacking*. Burlington: Syngress, 2004, s. 293–322, ISBN: 978-1-932266-83-2. DOI: 10.1016/B978-193226683-2/50015-6.
- [5] Game Verifying Team. "Visual guide to verifying N64 carts", viitattu 18. helmikuuta 2026. url: <https://gameverifying.com/wiki/cart-based-systems/n64>.
- [6] J. Ma, "From the Atari Shock to a Modern Crisis: Analyzing Mass Layoffs in the Post-Pandemic U.S. Video Game Industry", *CSCW '17*, 2025. DOI: 10.1145/3774758.

-
- [7] S. Arendt. "Civilization Creator Lists Three Most Important Innovations in Gaming", viitattu 18. helmikuuta 2026. url: <https://www.wired.com/2008/03/sid-meier-names/>.
- [8] shutterbug2000. "Memory Pit - A new DSi exploit for DSi camera!", viitattu 18. helmikuuta 2026. url: <https://gbatemp.net/threads/memory-pit-a-new-dsi-exploit-for-dsi-camera.539432/>.
- [9] NightScript, Pk11 ja Peter0x44. "DSi Guide", viitattu 18. helmikuuta 2026. url: <https://dsi.cfw.guide/launching-the-exploit.html>.
- [10] lifehackerhansol, lilyuwuu, eip618, ihaveamac, ja adrot. "3DS Hacks Guide", viitattu 24. maaliskuuta 2026. url: <https://3ds.hacks.guide/key-information.html>.
- [11] Nintendo Homebrew. "Wii Hacks Guide", viitattu 1. kesäkuuta 2026. url: <https://wii.hacks.guide/>.
- [12] Consoles Unleashed. "Nintendo Entertainment System Region Free Mod Install Guide", viitattu 24. maaliskuuta 2026. url: <https://www.consolesunleashed.com/guides/nintendo-entertainment-system-region-free-mod-install-guide/?v=75778bf8fde7>.
- [13] Yod@. "Modifying N64", viitattu 24. maaliskuuta 2026. url: <https://nfggames.com/forum2/index.php?topic=1523.0>.
- [14] Y. Bakrim. "PlayStation 2 ModChips", viitattu 24. maaliskuuta 2026. url: <https://altarofgaming.com/playstation-2-modchips/>.
- [15] G. Pessolano, H. O. Read, I. Sutherland ja K. Xynos, "Forensic Analysis of the Nintendo 3DS NAND", *Digital Investigation*, vol. 29, S61–S70, 2019, ISSN: 1742-2876. DOI: 10.1016/j.diin.2019.04.015.

-
- [16] Z. Skyforest. "Dumping Game Boy Cartridges Via The Link Cable Port", viitattu 5. toukokuuta 2026. url: <https://hackaday.com/2022/12/14/dumping-game-boy-cartridges-via-the-link-cable-port/>.
- [17] RocketRobz. "GodMode9i", viitattu 5. toukokuuta 2026. url: <https://wiki.ds-homebrew.com/godmode9i/>.
- [18] R. Copetti. "Nintendo 64 Architecture", viitattu 14. huhtikuuta 2025. url: <https://www.copetti.org/writings/consoles/nintendo-64/>.