

Tero Vaakanainen

ÄÄNIRAJAPINNAN TOTEUTUS KAJAK3D-PELIMOOTTORIIN

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Luonnontieteiden ala
Tietojenkäsittelyn koulutusohjelma
Syksy 2010



Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Tero Vaakanainen	
Työn nimi Äänirajapinnan toteutus Kajak3D-pelimoottoriin	
Vaihtoehtoiset ammattiopinnot Peliohjelmointi	Ohjaaja(t) Janne Koponen Toimeksiantaja Kajaanin ammattikorkeakoulu
Aika Syksy 2010	Sivumäärä ja liitteet 41 + 3
<p>Kajak3D-pelimoottori on usean alustan pelimoottori, joka toimii Windows-, Maemo- ja Iphone-alustoilla. Sen on kehittänyt Kajaanin ammattikorkeakoulun pelilaboratorio. Sitä jaetaan ilmaiseksi Kainuun-alueen yritysten käyttöön. Tämän opinnäytetyön tarkoituksena oli toteuttaa äänirajapinta Kajak3D-pelimoottoriin.</p> <p>Videopelit ovat kehittyneet videopelien 60-vuotisen historian aikana jokaisella osa-alueella. Interaktiivinen äänentoisto on vähitellen saavuttanut elokuvien tason myös matkapuhelin laitteissa. Äänentoistolaitteita ohjataan äänikirjastoilla. Äänikirjastot voivat olla mm. käyttöjärjestelmä tai laitekohtaisia. Videopelit sisältävät nykyään valtavan määrän yhtäaikaista mono- ja stereoääniä, jotka voidaan miksata reaaliajassa surround-ääniksi. Äänien toistaminen voi tapahtua puskuroimalla koko äänitiedosto kerralla tai streamaamalla eli puskuroimalla pieni osa tiedostoa toisto varten. Äänentoisto formaatin ja tiedoston toistotavan valinta riippuu alustasta ja käyttötilanteesta.</p> <p>Kajak3D-pelimoottori tukee tällä hetkellä Windows-, Maemo- ja iPhone-alustoja, sekä siihen kehitetään parhailaan tuki myös Android-alustalle. Tämän opinnäytetyön lopputuote Kajak3D-äänirajapinta tukee samoja alustoja, joskin Android-alustan tukeminen jää jatkokehityksestä vastaavien tahojen käsiin. Alustat eroavat toisistaan niin työkaluiltaan kuin käänösprosessiltaan.</p> <p>Äänirajapinnan toteutus alkoi asiakasvaatimuksesta, jonka pohjalta toteutin vaatimusmäärittelyn. Äänirajapinnan toiminnallisuus piti suunnitella vaatimusmäärittelyn pohjalta. Perustoteutus alkoi Windows alustalla. Kun perustoteutus oli valmis, äänirajapinnan kääntäminen muille alustoille alkoi. Käännösten valmistuttua oli vuorossa testaaminen, jossa vaatimusmäärittelyssä esitetyt kohdat testattiin. Äänirajapinnan rakenne on melko yksinkertainen. Sen käytössä on huomioitava tietty järjestys, että äänirajapinta toimii oikein. Äänirajapinnan käyttämät ulkoiset kirjastot toimivat tietyllä tapaa ja osa niistä on sidoksissa Kajak3D-pelimoottorin toiminnallisuuteen.</p> <p>Opinnäytetyön alussa käsiteltiin videopeliäänien historia ja nykypäivää, sekä digitaalisen äänentoiston teoriaa. Toteutusalustojen erot ja äänirajapinnan kääntäminen selitettiin kappaleessa 3 ja 4. Äänirajapinnan rakenne ja funktiot kerrottiin kappaleessa 5. Äänirajapinnan toteutus onnistui hyvin, mutta siinä on edelleen puutteita. Jatkokehityksessä se pitäisi kääntää Android-alustalle, sekä lisätä tuki muille tarvittavalle äänitiedostomuodoille. Äänirajapinnan perustoiminnallisuutta voisi myös virtaviivaistaa.</p>	
Kieli	Suomi
Asiasanat	Äänentoisto, pelit, peliäänet, äänirajapinta, kajak3d, videopelit, OpenAL, Iphone, Maemo
Säilytyspaikka	<input checked="" type="checkbox"/> Verkkokirjasto Theseus <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Kajaani University of Applied Sciences	Degree Programme Bachelor of Business Information Technology
Author(s) Tero Vaakanainen	
Title Audio Wrapper for Kajak3D Engine	
Optional Professional Studies Game programming	Instructor(s) Janne Koponen
	Commissioned by Kajaani University of Applied Sciences
Date Fall 2010	Total Number of Pages and Appendices 41 + 3
<p>The Kajak3D engine is a multiplatform game engine made by the Kajak Game Development Laboratory of the Kajaani University of Applied Sciences. Currently Kajak3D works on Windows, Maemo and Iphone platforms. It is freely distributed to businesses in the Kainuu area. The purpose of this thesis was to produce the audio wrapper for the Kajak3D engine.</p> <p>Video games have evolved during their almost 60-year history in every sector. Interactive audio has reached movie quality even on handheld devices. Audio devices are controlled with audio libraries and those libraries can be limited to one operating system or device. Games today have multiple mono and stereo tracks running at the same time, and the output can be mixed for a full surround playback in real time. The sound files can be buffered fully or streamed depending on the capability of the playback device.</p> <p>Kajak3D works currently on Windows, Maemo and Iphone platforms and the support for the Android platform is under development. The end product of this thesis works in the same platforms, but the support for Android is not part of this thesis. Operating systems differ greatly from each other and the tools and the development process are different.</p> <p>The development of the audio library started from making a customer specification. Requirement specification was made using the customer specification. After the requirements were clear, the design of the audio library could be started. The basic functionality was made for Windows. After that, the porting for the other platforms could be started. After the software worked on all the platforms, it was time to start the testing, in which all the requirements were tested. The structure of the wrapper is pretty simple, but there is a certain order in which the function calls must be made for it to work correctly. The external libraries that the wrapper uses also work in certain ways and some of them depend on the functionality of the Kajak3D engine.</p> <p>The thesis started from the history of game sounds and explained the basic principles of digital audio. Platforms and their differences were explained in general. The thesis also contains the development process and the basic structure of the wrapper. There are some things to improve in the audio wrapper. Support for Android and other new platforms needs to be added. Also, some changes could be made to streamline the usage of the wrapper.</p>	
Language of Thesis	Finnish
Keywords	audio, sounds, Games, video games, Kajak3D, OpenAL, iPhone, Maemo, audio wrapper
Deposited at	<input checked="" type="checkbox"/> Electronic library Theseus <input checked="" type="checkbox"/> Library of Kajaani University of Applied Sciences

SISÄLLYS

SYMBOLILUETTELO

1 JOHDANTO	1
2 PELIÄÄNET	2
2.1 Historiaa	3
2.1.1 Ensimmäiset pelit ja pelihallien aika	3
2.1.2 Matkapuhelimet ja mobiilipelit	5
2.1.3 Käsikonsolit	7
2.1.4 Kotitietokoneet ja PC	9
2.1.5 Kotikonsolit	11
2.2 Nykypäivä	15
2.3 Äänentoisto	17
2.3.1 Mono	17
2.3.2 Stereo	17
2.3.3 Surround	18
2.4 Äänten toistaminen	19
3 ALUSTAT	21
3.1 Windows	21
3.2 iPhone	21
3.3 Maemo	22
4 ÄÄNIKIRJASTON TOTEUTUS	23
4.1 Vaatimusmäärittely ja suunnittelu	23
4.2 Perustoteutus	23
4.3 Maemo-käännös	24
4.3.1 Maemo-asetukset	24
4.3.2 Scratchbox-asetukset	25
4.3.3 Kääntäminen	25
4.4 Iphone käännös	26
4.5 Testaus	28
5 KAJAK3D-ÄÄNIRAJAPINNAN RAKENNE	29
5.1 Äänirajapinnan rakenne	29

5.1.1 Soundmanager	29
5.1.2 Audiostream	30
5.1.3 Oggstream	30
5.1.4 Wavstream	30
5.2 Äänilaitteen alustus	30
5.3 Tiedoston luku	31
5.3.1 Wav	32
5.3.2 Ogg	33
5.4 Äänitiedoston toistaminen	33
5.4.1 Mono- ja stereo-raidat	34
5.4.2 Etäisyys ja efektit	34
5.4.3 Streamaus ja kokotiedostontoisto	34
5.5 Äänien päivitys ja hallinta	35
6 YHTEENVETO	36
LÄHTEET	37
LIITTEET	

SYMBOLILUETTELO

ALUSTA Platform eli ohjelmisto, laite, käyttöjärjestelmä tai näiden yhdistelmä, jolla ohjelma, ohjelmisto tai kirjasto toimii tai on tarkoitettu toimivaksi.

CORE AUDIO Matalan tason rajapinta äänenkäsittelyyn Iphone OS ja Mac OS X käyttöjärjestelmissä

IPHONE OS Iphone Operating System on Apple yhtiön iPhone, iPad ja iPod mobiililaitteisiin suunniteltu käyttöjärjestelmä ja sen eri versiot toimivat näissä laitteissa

MAEMO Linux-pohjainen käyttöjärjestelmä Nokian N900 mobiililaitteille

MIDI Musical Instrument Digital Interface eli musiikkisoittimen digitaalinen rajapinta, joka on suunniteltu tiedonvälitykseen sähköisten musiikkilaitteiden välillä. MIDI:llä viitataan myös MIDI-äänitiedostoihin, jotka pitävät sisällään vain musiikin digitaaliset soitto-ohjeet eli nuotit, eivätkä varsinaisia ääniä, jotka soitetaan laitteen omalla MIDI-syntetisaattorilla. Tiedosto pääte ".mid"

NPC eli Non Playable Character on videopeleissä esiintyvä pelihahmo, johon pelaajalle ei ole suoraa kontrollia, vaan pelaaja joutuu suorittamaan interaktioita niiden kanssa esimerkiksi ampumalla niitä, keskustelupuulla tai muilla tavoin

OGG VORBIS Ogg Vorbis on lisenssimaksuton äänenpakkausmenetelmä, joka pohjautuu avoimeen lähdekoodiin. Tiedosto pääte ".ogg"

OPEN AL Open Audio Library on usean alustan 3D äänikirjasto ja rajapinta, joka on kehitetty erityisesti käytettäväksi peleissä, sekä erilaisissa muissa äänisovelluksissa. äänirajapinta mallintaa usean eri äänilähteen liikkumista 3D avaruudessa yhden kuuntelija näkökulmasta

PCM Pulse Code Modulation eli pulssikoodimodulaatio on menetelmä, jolla sähköinen äänitaajuussignaali koodataan digitaaliseen muotoon. Siinä analogisesta signaalista otetaan näytteitä tasaisin väliajoin ja ne ilmaistaan numerisesti. Näytteen taso voidaan ilmaista joko lineaarisesti tai logaritmisesti

PELIMOOTTORI Pelimoottori on videopelien kehittämiseen tehty ohjelmisto, joka sisältää yleensä vähintään grafiikan esittämiseen ja 3d-objektien hallintaan tarvittavat kirjastot.

Pelimoottorit voivat myös sisältää esimerkiksi äänten toistamiseen, scriptaukseen, animaatioiden käsittelyyn ja verkkopelaamiseen tarvittavia kirjastoja

PSG Programmable sound generator eli ohjelmoitava äänigeneraattori. Äänisiru joka generoi ääniaaltoja useasta perusäänialto muodosta.

SIIRRETTÄVYYS Portability eli kuinka helppo tai pieni on tietynohjelman tai ohjelmiston muokkaustarve eri alustalta siirryttäessä toiselle. Ideaali tilanne on, ettei ohjelmiston lähdekoodiin tarvitse tehdä muutoksia.

SURROUND eli monikanavaisesta äänentoistosta käytetty termi, jossa usealla kaiuttimella kuuntelijan ympärille luodaan kuva siitä mistä päin äänet kuuluvat

WAV Wav tai Wave eli Waveform Audio File Format on yleisin käytössä oleva äänenpakkaus formaatti, joka voi sisältää pakattua tai pakkaamatonta äänitietoa. Tiedostopäätte Windows alustalla ".wav"

WINDOWS Microsoftin käyttöjärjestelmä, jonka uusin versio on Windows 7

1 JOHDANTO

Pelejä kehitettiin vielä 2000-luvun puolivälissä pienillä tiimeillä, nykyään pelejä tehdään jopa 300 hengen tiimeillä. Tämä johtuu siitä, että peleissä on parhaillaan useiden animaatioelokuvien verran välivideoita, ääniä, sekä satojen neliökilometrien kokoisia pelimaailmoja. Suuret pelimaailmat sisältävät yleensä useita kymmeniä tuhansia erilaisia 3D-malleja ja saman verran tekstuureja. Välivideot taas voivat vielä parhaillaan olla interaktiivisia, joissa pelihahmo puhuu pelaajan valintojen mukaan. Nämä valinnat johtavat siihen, että pelaajahahmo voi olla esimerkiksi nainen tai mies, sekä puhua tietyn keskustelun pätkän vihaisella, iloisella, neutraalilla tai uhkaavalla tavalla. Tämä tarkoittaa sitä, että ääninäyttely pitää suorittaa useaan otteeseen jokaisella osa-alueella, jossa pelaajan toiminta vaikuttaa pelihahmon puhuttuun dialogiin. Suuret tiimit sisältävät tuottajia, käsikirjoittajia, mallintajia, pelisuunnittelijoita, graafikoita, säveltäjiä, äänisuunnittelijoita, ohjelmoijia, testaajia, liikkeenkaappausnäyttelijöitä ja tähän ei vielä lasketa kustantajan puolella projektissa mukana olevia henkilöitä.

Tämä on johtanut siihen, että pienellä tiimillä ei nykyään edes pyritä tekemään isoja peliprojekteja. Kuitenkin pienille tiimeille on muita keinoja tehdä hyvä laatuista pelejä, jotka eivät tarvitse miljoonaluokan budjetteja valmistuakseen. Erityisesti mobiilipelit, konsolien ja pc:n latauspalvelut ovat kehittyneet suunnattomasti viime vuosina. Latauspalvelujen etuna on alhainen aloituskynnys ja helppous saada valmis peli kuluttajille.

Latauspalvelut ovat aloittelevalla peliyritykselle hyvä paikka aloittaa toimintansa, juuri yllämainituista syistä. Tyypillisen aloittelevan peliyrityksen ensimmäinen tehtävä on hankkia teknologia, jolla peli kehitetään eli pelimoottori. Tämä vie aikaa ja rahaa, joten Kainuun alueelle perustettavia ja perustettuja yrityksiä varten, jotka tarvitsevat pelimoottoritekniologiaa, on kehitetty Kajak3D-pelimoottori. Kajak3D-pelimoottori on Kajaanin ammattikorkeakoulun pelinkehityslaboratorion kehittämä pelimoottori, joka toimii tätä kirjoittaessa Windows, Iphone ja Maemo alustoilla. Tätä pelimoottoria jaetaan ilmaiseksi Kainuun alueen yrityksille.

Kajak3D-pelimoottori on osa SIMUPELI-hanketta. SIMUPELI-hankkeen on tarkoitus luoda peliteollisuutta ja simulaatiotekniologia-yrityksiä Kainuun alueelle ja tarjota niille hyvät olosuhteet kasvuun ja vakaaseen liiketoimintaan.

Kajak3D-pelimoottorissa ei vielä ole äänirajapintaa, joka toimisi kaikilla kolmella tuetulla alustalla. Erillisiä äänikirjastoja voidaan käyttää kaikilla alustoilla, joita Kajak3D tukee, mutta

opiskelijan tai kehittäjän resurssit eivät välttämättä riitä toteuttamaan ääniä kaikille kolmelle alustalle. Pelinkehittäjän kannalta yhtenäinen siirrettävä äänikirjasto, joka toimii kaikilla kolmella alustalla, ratkaisee monta ongelmaa ja helpottaa heidän kehitystyötään. Mikäli pelikehittäjä käyttäisi erillisiä äänikirjastoja, niiden toiminnallisuus ja dokumentointi on erilainen. Se aiheuttaisi lisää työtä, johon kehittäjällä ei välttämättä riitä resurssit.

2 PELIÄÄNET

Tietokoneet, pelikonsolit ja puhelimet ovat kehittyneet valtavasti lyhyessä ajassa. Samalla näiden laitteiden äänentoisto on kehittynyt valtavan nopeasti ja äänten suunnittelusta on tullut haastavampaa ja monimutkaisempaa. Alun yksikanavaisista piippauksista ollaan tultu jopa 8-kanavaiseen surround-äänentoistoon. Siinä missä ennen pelimusiikki tehtiin yhden ohjelmoijan tai muusikon toimesta, nykyään kappale voidaan esittää jopa kokonaisen sinfoniaorkesterin voimin. Erikoisempia ratkaisuja kuten se, että musiikki generoidaan pelaajan tekojen mukaan, on myös käytetty useassa pelissä. Kehitystä ei ole tapahtunut vain taustamusiikin suhteen, sillä myös äänitehosteet ovat tulleet piippauksista siihen, että pelit käyttävät oikeita ääniä niiden tilalla. Äänitehosteet ovat usein samojen äänistudioiden tekemiä, jotka tekevät äänet Hollywood-elokuviin. Myös useat Hollywood-näyttelijättäret ja -näyttelijät ovat lainanneet äänensä pelihahmoille. Kappaleessa 2.1 käsitellään videopelien ja peliäänten kannalta olennainen historia, niin matkapuhelinten ja tietokoneiden, kuin pelihallien ja kotikonsolien osalta.

Nykyiset kotikonsolit, käsikonsolit, puhelimet ja tietokoneet ovat osa jokapäiväistä viihdettä. Ne ovat valloittaneet niin vapaa- kuin työajankin. Kappaleessa 2.2 kerrotaan peliteollisuuden nykyisestä tilanteesta.

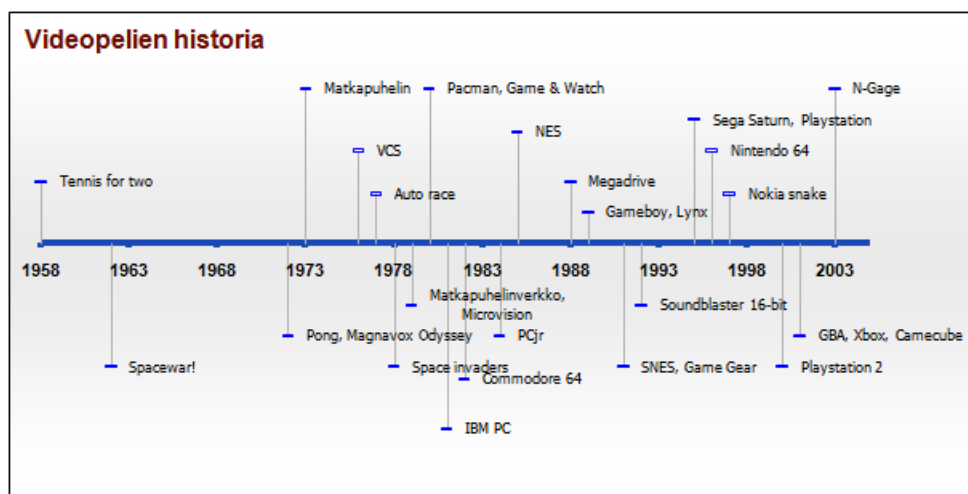
Äänten toistamiseen, äänitiedostojen tallentamiseen, äänittämiseen ja käsittelyyn on useita eri tapoja, tiedostomuotoja ja standardeja. Mono-, stereo- ja surround-äänentoistolla on kaikilla yhteisiä piirteitä, mutta ne myös eroavat toisistaan. Tämän kappale 2.3 kertoo näiden toistomahdollisuuksien eroista ja erityispiirteistä.

Äänten käsittely tietokoneella on peliohjelmoijan kannalta helppo jakaa kolmeen yleiseen vaiheeseen, jotka ovat äänitiedoston lukeminen, luetun tiedon purkaminen ja puretun tiedon toistaminen. Näissä vaiheissa täytyy ottaa huomioon resurssit (tässä kappaleessa prosessoriteho, käyttömuisti, tallennustila), jotka laiteella on käytössään. Näihin asioihin paneudutaan tarkemmin kappaleessa 2.4, joka on hyödyksi erityisesti laitteille, joissa on vähän laskentatehoa eli mobiililaitteet.

2.1 Historiaa

Pelit ovat kehittyneet vähän yli 50-vuotiaan historiansa aikana pienen piirin harrastuksesta suurimmaksi viihdeteollisuuden haaraksi. Kaikki alkoi yliopistoista, siirtyi pelihalleihin, ja lopulta valloitti aikamme olohuoneissa, matkalla ja töissä.

Kuvassa 1 on tämän kappaleen 2.1 kannalta esitetyt asiat tiivistetyllä aikajanalla. Videopelissä on myös tapahtunut paljon muuta, mutta aikajanalla olevat tapahtumat ovat ehkä merkityksellisimmät videopeliänten kannalta.

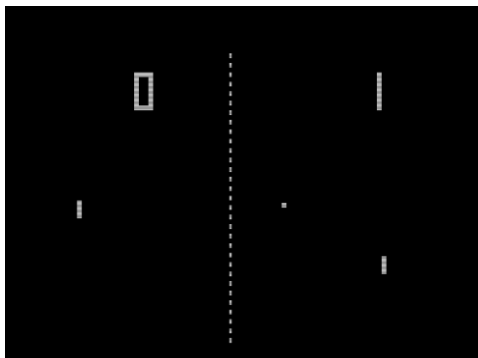


Kuva 1. Videopelien historia äänten kannalta esitettynä aikajanalla

2.1.1 Ensimmäiset pelit ja pelihallien aika

Ensimmäiset videopelit olivat äänettömiä. Näihin voidaan laskea William Higginbothamin Tennis for Two vuonna 1958 ja Steve Russellin Spacewar! vuonna 1962. Nämä pelit käyttivät näyttöinänsä oskilloskooppia, joskin Spacewar! käännettiin myös muille sen aikaisille tietokoneille, joista osa tuki näyttöä. Myös ensimmäiset kotikonsolit olivat äänettömiä, kuten Magnavox Odyssey 1972. (Collins, K. 2)

Ensimmäinen peli jossa oli äänet, oli Atarin Pong (kuva 2.) vuonna 1972. Pelissä oli ainoastaan muutama piippaus, joka kuului pallon osuessa mailaan, seinään tai kun tehtiin piste. Atarin Pong oli suuri hitti silloisissa pelihalleissa. Pong aloitti silloisten pelihallien siirtymän flippiautomaateista elektronisiin videopeleihin. (Collins, K. 2)



Kuva 2. Atarin Pong peli (Hill, J.)

Ensimmäinen arcadepelejä joka esitteli taustamusiikin, oli Taito Midwayn Space Invaders (kuva 3.) vuonna 1978. Sen taustamusiikki oli neljä peräkkäistä nuottia, jotka kiihtyivät sitä mukaa, mitä lähemmäksi avaruusolot tulivat pelaajan tukikohtaa. Tämän lisäksi Space Invaders sisälsi 6 ääniefektiä. Merkittävin asia oli kuitenkin taustamusiikin interaktiivisuus eli pelin taustamusiikin tempon kiihtyminen vihollisalien nopeuden mukaan. Tämä loi pelaajalle tietynlaisen vaarantunteen pelin edetessä. (Collins, K. 2)



Kuva 3. Taito Midwayn Space Invaders peli ja kabinetti (Andrews, S)

Suurin osa ensimmäisistä arcadepeleistä sisälsi yhden tai kahden kanavan melodian, joko alun tunnusmusiikkina tai muutaman sekunnin pituisena pelimusiikkisilmukkana, jota toistettiin koko pelin ajan. Tähän tuli kuitenkin muutos, kun peleissä alettiin käyttää välivideoita. Välivideoiden käyttäminen mahdollisti monimutkaisemman musiikin esittämisen, koska välivideoiden aikana prosessointitehoa ei mennyt pelaajan kontrollien lukemiseen, eikä vihollisten liikkeen laskemiseen. Ensimmäinen välivideo nähtiin vuonna 1980 Tohru Iwatanin Pacman (kuva 4.) pelissä. Pelissä kuultiin siihen aikaan pitkä ja monimutkainen pätkä musiikkia tämän välivideon aikana. (Collins, K. 2)



Kuva 4. Pacman peli (Wired.com) ja videopelikabinetti (Arcadeshop.com)

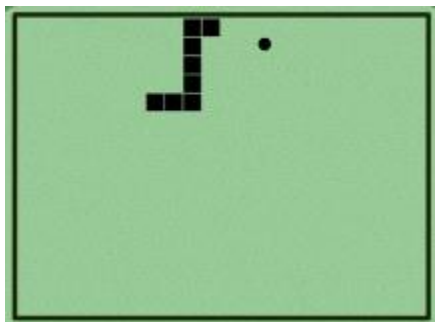
Pelihallit ja videopelikabinetit olivat tekemässä videopeleistä suosittua koko kansan huvia, mutta ne ovat pikkuhiljaa poistumassa. Nopeat internetyhteydet ja konsolipelaamisen helppous ovat tehneet pelihalleista länsimaissa kuolevan ilmiön. Japanissa pelihallikulttuuri on edelleen voimissaan, joskin sielläkin sen suosio on hiipumassa.

2.1.2 Matkapuhelimet ja mobiilipelit

Samoihin aikoihin, kun Pong valloitti pelihallit, puhelinvalmistajat Bell Labs ja Motorola kilpailivat siitä kumpi keksisi ajoneuvoista riippumattoman matkapuhelimen. Vuonna 1973 Martin Cooper, joka toimituskijana Motorolalla teki ensimmäisen puhelun nykyaikaisella matkapuhelimella kilpailijalleen Joel S. Engelille, joka työskenteli Bell Labsilla. (Messmer, E.)

Ensimmäinen kaupallinen automatisoitu matkapuhelinverkko aloitti toimintansa Japanissa Tokion ydinkeskustassa vuonna 1979. (Jeffs, A) Pohjoismaiden verkko NMT-450 aloitti toimintansa vuonna 1981 Ruotsissa, Tanskassa, Norjassa, sekä Suomessa vuotta myöhemmin 1982. Se oli maailman ensimmäinen täysin automatisoitu kaupallinen matkapuhelin-verkko. (Juutilainen, M. 9)

Ensimmäinen kännykkäpeli Snake (kuva 5.) julkaistiin vuonna 1997, jolloin se oli esiasennettuna tiettyihin Nokian matkapuhelin malleihin. Silloiset puhelimet tukivat yksikanavaista piippausääntä eri korkeudelle ja pituudella. Vaikka peli olikin yksinertainen se toimi lähtökohtana ja innoittajana kokonaisen uuden teknologia-alan kehitykselle, joka on johtanut nykyiseen tilanteeseen, jossa mobiilipelit ovat miljardiluokan bisnes. (Paavilainen, Korhonen & Saarenpää.)



Kuva 5. Snake (Torumi.com)

Samoin kuin PC-, konsoli- ja arcadepeleissä, kännyköiden äänet kehittyivät seuraavaksi tukemaan midi-ääniä. Ensimmäinen midi-soittoääniä tukeva puhelin julkaistiin Japanissa jo vuonna 1996. (Guerrieri, M.)

Midi-äänien aika jäi kuitenkin lyhyt aikaiseksi ilmiöksi kännyköissä, sillä kannettavien mp3-soittinten yleistymisen johti siihen, että kännykkävalmistajat integroivat tämän ominaisuuden nopeasti puhelimiinsa. Ensimmäinen puhelin, jossa oli tuki mp3-tiedostoille julkaistiin vuonna 1999 Etelä-Koreassa. Puhelin oli Samsungin valmistama. (Samsung)



Kuva 6. Nokian N-Gage puhumisasennossa (Mobile-review.com)

Nokia julkaisi pelaajille suunnatun puhelimensa N-Gagen vuonna 2003. N-Gage oli matkapuhelimen ja käsikonsolin risteytys. Se tuki, myös mp3-tiedostojen toistoa ja oli samalla kannettava mp3-soitin. Pelit toimitettiin muistikorteilla ja puhelin täytyi sulkea vaihtaakseen pelin. Pelipuhelin sai huonon vastaanoton niin pelaajilta kuin puhelinten käyttäjiltä. Se, että pelin vaihtaakseen puhelin piti sulkea ja takakansi avata ei ollut käyttäjistä

hyvä ratkaisu, eikä myöskään omalaatuinen mikrofonien sijoittelu auttanut asiaa (kuva 6). (Paavilainen, Korhonen & Saarenpää.)

Nokian N-gage oli ensimmäinen kunnan yritys puhelinvalmistajalta tunkeutua käsikonsolimarkkinoille, mutta läpimurto tapahtui vuosia myöhemmin Applen Iphonen muodossa. Applea alettiin puhelimen suosion myötä pitää vakavasti otettavana peliyhtiönä vaikka se ei aktiivisesti pyrkinyt olemaan sellainen. (Shcramm, M)

2.1.3 Käsikonsolit

Ensimmäinen elektroninen kannettava pelilaitte oli Mattelin Auto Race (kuva 7.) vuonna 1977. Auto Race sisälsi äänet auton moottorille ja törmäykselle. Auton vaihdesäädin säätö samalla myös moottorin äänennopeutta vaihteen yksi ollessa hidasta ja vaihteen neljä ollessa nopea. (Barton & Loguidice.)

Ensimmäinen kannettava pelilaitte johon pystyi vaihtamaan pelikasetin, oli Milton Bradleyn Microvision. Microvision ilmestyi vuonna 1979. Microvisionin äänet olivat perinteisiä piip-paus-ääniä. Pelikasetit olivat samalla laitteen etuosa ja sisälsivät pelin tarvitsemat napit (kuva 7.). Ainut käsikonsolissa valmiina oleva ohjain oli rullasäädin. (Melanson, D.)

Nintendon menestys pelihalleissa, johti siihen, että Nintendo alkoi pikkuhiljaa siirtyä videopelien maailmaan. Ensimmäinen Nintendo Game & Watch -sarjan peli Ball (kuva 7.) ilmestyi vuonna 1980. Game & Watch -sarjan idea oli, että sen laitteet olivat kellon ja elektronisen videopelin yhdistelmä. (Melanson, D.)



Kuva 7. Mattelin Auto Race (Beeslife.com), Milton Bradley Microvision (Handheldmuseum.com) ja Nintendon Game & Watch Ball (Sobcontrollers.com)

Nintendo astui käsikonsolien maailmaan Game Boy -konsolilla vuonna 1989 (kuva 8.). Game Boysta tuli ensimmäinen supersuosittu käsikonsoli. Nintendon aikaisempi menestys pelikonsolimarkkinoilla auttoi kuluttaja tietoisuudessa, mutta suurin myyntivaltti oli kuitenkin siihen aikaan supersuosittu Tetris. Nintendo Game Boy sisälsi nelivärisen harmaasävynäytön, joka mahdollisti paristojen pidemmän eliniän. NES-konsolin tavoin Game Boy ei sisältänyt erillistä äänipiiriä vaan se oli integroitu prosessorin yhteyteen. Game Boy:n äänentoisto-ominaisuudet vastaavat lähes NES-konsolin ominaisuuksia. (Melanson, D.)

Game Boy -konsolia seurasi pienempi versio Game Boy Pocket (kuva 8.) ja värinäytöllinen Game Boy Color (kuva 8.). Game Boy Color oli taaksepäin yhteensopiva ja osa sen peleistä oli taaksepäin yhteensopivia alkuperäisen Game Boy:n kanssa. (Melanson, D.)



Kuva 8. Game Boy (Ratteswg.de), Game Boy Pocket (Rolentapress.com), Game Boy Color (Pauljamesmanning.com)

Game Boy:n suosion takia käsikonsolimarkkinoita tavoittelivat myös Atari ja Sega. Vaikka Atari menestyi melko hyvin kotikonsolimarkkinoilla, vaikutti vuoden 1983 suuri videopeli-markkinoiden romahdus myös sen toimintaan. Atari yritti käsikonsolimarkkinoille sen kannettavalla pelikonsolillaan Atari Lynx (kuva 9.). Lynx ilmestyi samana vuonna kuin Game Boy eli 1989. Nintendon konsoli oli halvempi noin 90 dollarin hinnalla ja Atarilla oli ongelmia laitetoimitusten kanssa joulumarkkinoilla. Niinpä Nintendo sai markkinajohtajuuden. (Melanson, D.)



Kuva 9. Sega Game Gear (gamesetwatch.com) ja Atari Lynx (wikipedia.fi)

Sega, joka oli Nintendon ainut varteenotettava kilpailija konsolimarkkinoilla pitkään aikaan, julkaisi oman käsikonsolinsa Game Gearin (kuva 9.) vuonna 1991. Koska Game Gear perustui Sega Master System -konsoliin, pystyi Sega nopeasti luomaan vanhoista peleistään uusintaversioita tukeakseen käsikonsoliaan. Vaikkei Game Gear koskaan saavuttanut samanlaista suosiota kuin Game Boy, oli se kuitenkin pitkäikäisin kilpailija, joka Nintendolla koskaan on ollut käsikonsolimarkkinoilla. (Melanson, D.)



Kuva 10. Game Boy Advance (Pugo.org), Game Boy Advance SP (Rankomedia.com) ja Game Boy Advance Micro (Swotti.com)

Nintendon viimeinen kehitysversio Game Boy -tuotesarjaan oli Game Boy Advance (kuva 10.), joka oli tehokkaampi kuin alkuperäinen Game Boy ja vastasi teholtaan suurin piirtein SNES- eli Super Nintendo -konsolia. GBA julkaistiin vuonna 2001. Siitä julkaistiin kaksi lisäversiota GBA SP (kuva 10.) ja GBA Micro (kuva 10.), jotka molemmat olivat pienempiä kuin alkuperäinen ja toivat parannuksia sekä näyttöön että akkukestoon. (Melanson, D.)

Käsikonsolimarkkinat ovat laitemyyneissä suuremmat kuin kotikonsolimarkkinat. Niiden etuna ovat olleet halvempi hinta, mukana kulkevuus ja se etteivät käsikonsolit tarvitse erillisiä lisälaitteita toimiakseen. Pelit ovat käsikonsoleille halvempia, mutta niiden pelimyyntit eivät ole yhtä suuret kuin kotikonsoleiden. Käsikonsoleiden aika nykymuodossaan alkaa olla takanapäin, sillä matkapuhelimet ja mp3-soittimet ovat alkaneet tunkeutua niiden markkina-alueelle.

2.1.4 Kotitietokoneet ja PC

Vuonna 1981 IBM esitteli ensimmäisen PC-tietokoneensa (kuva 11.). Tämä tapahtui vuonna 1981. Ensimmäiset IBM PC-tietokoneet ja niiden kloonit sisälsivät ainoastaan pienen kaiuttimen, joka pystyi tuottamaan yksinkertaisia ääniä eri taajuuksilla, mutta aina samalla voimakkuudella. Kaiutinta käytettiin lähinnä ilmoittamaan tietokoneen virheilmoituksia erilaisilla merkkäänillä. (Collins, K. 2)

Vuonna 1982 Commodore Business Machines yhtiö esitteli uuden kotitietokoneensa Commodoren 64:n (kuva 11.), jota myytiin maailmanlaajuisesti n. 22 miljoonaa kappaletta. Commodore on peliäänten kehityksen kannalta merkityksellinen, koska se sisälsi SID (Sound Interface Device) äänipiirin. Sen äänentoistokyky oli siihen aikaan poikkeuksellisen korkea ja alhainen 200 dollarin hinta kotitietokoneesta kuluttajille houkutteleva. (Collins, K. 2)

IBM:n seuraavan sukupolven tietokone PCjr (kuva 11.) ilmestyi 1984. Se sisälsi parannetut ääni- ja grafiikkaominaisuudet, joiden tärkeyden IBM ymmärsi Commodoren ja Applen menestyksestä kotitietokonemarkkinoilla. Markkinoidakseen tietokoneen uusia kehittyneitä ominaisuuksia IBM palkkasi Sierra Entertainmentin tekemään pelin PCjr alustalle, jonka tarkoituksena oli ottaa kaikki irti parantuneesta suorituskyvystä. Peli, jonka Sierra toteutti, oli King's Quest, ensimmäinen graafinen "3D"-seikkailupeli. PCjr ei kuitenkaan menestynyt kovin hyvin, mutta sen samaa tekniikkaa käyttävistä klooneista tuli melko suosittuja. (Collins, K. 2)



Kuva 11. IBM-PC (Academic.ru), Commodore 64 (Crackerwax.com) ja IBM PCjr (Old-computer.com)

PC-formaatin avoimuudesta johtuen piirejä valmistavat tahot alkoivat ymmärtää, että PC-pelaajat ja muusikot halusivat siedettävät ääniominaisuudet ilman, että heidän täytyi ostaa kokonaan uutta tietokonetta. Kolmannen osapuolen FM-äänikortteja alkoi ilmestyä markkinoille 1980-luvun puolivälissä. Nämä äänikortit oli suunniteltu pelaajat mielessä pitäen ja sisälsivät niin kutsutun peliportin johon voitiin liittää eri peliohjaimia. (Collins, K. 2)

Aiemmin pelien äänenlaatu riippui äänikortista, sillä midi-standardi määrittää, mikä soitin sijaitsee missäkin virtuaali-äänipankissa. Midi-standardi ei se kuitenkaan määritä millainen ääni eri äänikorteilla, esimerkiksi basso on. Säveltäjä ei siis voinut tietää, miltä pelin äänet kuulostivat eri laitteistolla. Tämä muuttui kuitenkin CD-levyformaatin yleistyttyä tietokoneissa. PC-pelit alkoivat pian ilmestyä CD-ROM-levyille. CD-levyn suurempi tallennuskapasiteetti

teetti mahdollisti parempilaatuisen musiikin ja grafiikan lisäämisen peleihin, koska tallennusformaatin koko ei ollut enää ongelma. (Collins, K. 2)

1990-luvulla tietokoneiden äänet ottivat suurimman harppauksensa Soundblaster-äänikorttien tullessa markkinoille. Ne tukivat PCM-ääniä ja mahdollistivat CD-tasoisien ääntöisto tietokoneella. Ne sisälsivät myös FM-piirin ja midi-tuen. (Collins, K. 2)

2.1.5 Kotikonsolit

Kotikonsoleiden taival alkoi vuonna 1972, kun Magnavox julkaisi ensimmäisen videopelikonsolin, joka voitiin liittää televisioon. Konsolissa oli 12 esiohjelmoitua peliä ja pelikasetit eivät sisältäneet muuta kuin jumpperit, joilla tietyt piirit laitettiin yhteen, jolloin tietty peli aktivoitui. Magnavox Odyssey (kuva 12.) ei sisältänyt ääniä. (Winter, D.)

Pong-pelin saatua suurta suosiota pelihalleissa, pyrki Magnavox hyödyntämään sen suosiota julkaisemalla uuden konsolin Odyssey 100:n. Odysseys 100 ei käyttänyt vaihdettavia pelikasetteja, vaan se sisälsi kaksi sisään rakennettua peliä Tenniksen (Pong) ja jääkiekon. Odyssey 100 oli ensimmäisiä Pong-konsoleita ja ensimmäinen kotikonsoli, joka sisälsi äänet. (Winter, D.)

Atarin Pongin oltua valtava hitti, myös Atari siirtyi tekemään kotikonsoleita. Atari julkaisi ensimmäisen konsolinsa Video Computer System:in (kuva 12.) 1976, jota myöhemmin kutsuttiin Atari 2600:si. Vuoden 1977 videopelimarkkinoiden romahduksesta Atari selvisi lähinnä, koska se oli juuri kääntänyt Space Invaders -pelin konsolilleen. Space Invaders oli niin suosittu, että useat ihmiset ostivat konsoleitaan vain sen takia. (Collins, K. 2)



Kuva 12. Magnavox Odyssey (Videogameconsolelibrary.com) ja Atari VCS (Scott, J)

8-bittisten pelikonsolien ehdoton kuningas oli kuitenkin Nintendo Entertainment System eli NES (kuva 13.), joka julkaistiin Amerikassa 1985. Konsoli julkaistiin Nintendo Family Computer eli Famicon nimellä Japanissa jo vuonna 1983. NES:in ostajat saivat konsolin mukana Super Mario Brothers -pelin, jonka musiikit ovat nykypäivänä klassikon asemassa. NES-konsolista kasvoi ajan mittaan todella suuri hitti ja se sai Nintendon siirtymään kokonaan pelikonsolibisnekseen. (Collins, K. 2)

Nintendon konsoli käytti prosessoriin integroitua viisikanavaista äänipiiriä. Äänipiirin kanavat tukivat jokainen yhtä ääniaaltomuotoa. Kaksi kanavaa oli varattu pulssiaalloille, yksi kolmioalloille, yksi kohinalle ja yksi sampleille. Pulssiaaltokanavat pystyivät toistamaan kahdeksan eri oktaavia ja niiden äänenvoimakkuutta pystyi säätämään 16 eri tasolle. Kolmioaalto-kanava sisälsi kiinteän äänenvoimakkuuden ja 16 esiohjelmoitua taajuutta. Sample-kanava oli niin sanottu DPCM-kanava eli Differential Pulse Code Modulation kanava. Se pystyi toistamaan myös normaalia PCM-ääntä. PCM-ääntä ei kuitenkaan käytetty kuin muutamassa pelissä pelikasettien tilanpuutteen vuoksi. (Collins, K. 2)

Nintendon suurin kilpailija 1990-luvun alussa oli Sega, jonka menestynein konsoli Mega Drive (kuva 13.) ilmestyi Japanissa vuonna 1988, Pohjois-Amerikassa 1989 ja Euroopassa 1990. Sega oli kaikin puolin tehokkaampi konsoli kuin NES ja siinä oli kaksi äänipiiriä ja kuusi kanavainen FM/PCM-piiri ja neljä kanavainen PSG-piiri. (Collins, K. 2)

Koska Mega Drive oli tehokkaampi kuin NES, Nintendo tajusi, että sen täytyisi julkaista oma 16-bittinen konsolinsa kilpaillakseen Mega Driven kanssa. Nintendon 16-bittinen konsoli Super Nintendo eli SNES (kuva 13.) ilmestyi vuonna 1991. Se sisälsi huomattavasti tehokkaammat grafiikka- ja äänipiirit. SNES:n äänipiiri sisälsi kahdeksan kanavaa ja sen ulostulo miksattiin 32 kilohertsin 16-bittiseksi ääneksi (CD-ääni on 44.1 kilohertsia 16-bittisenä). (Collins, K. 2)



Kuva 13. Nintendo NES (Pugo.org), Sega Mega Drive (Retrogamer.ch) ja SNES (Surrealist.de)

Sega Saturn (kuva 14.) julkaistiin vuonna 1995. Kuten kilpailijansa Sony, myös Sega käytti CD-formaattia konsolinsa tallennusformaattina. Sega Saturn oli ensimmäinen konsoli, joka hyödynsi internet-yhteyttä. Sega Saturnin suurin ongelma oli kuitenkin se, että se sisälsi paljon erilaista laitteistoa ja oli vaikea ohjelmoida, joten halvempi ja paremmin tuettu Playstation on saavutti suuremman suosion. (Collins, K. 2)

Sony Playstation (kuva 14.) ilmestyi vuonna 1995. Playstation suurin vaikutus on, että se sai uuden sukupolven pelaamaan. Playstation käytti CD-levyjä peliformaattinaan ja pystyi toistamaan myös normaaleja CD-levyjä, sekä CDV eli CD-videolevyjä. Playstationin tarina alkoi itse asiassa CD-levy laajenuksena Nintendon SNES-konsoliin, joka Sonyn oli tarkoitus kehittää yhteistyössä Nintendon kanssa. Nintendo kuitenkin purki sopimuksen ja julkisti yhteistyön Philipsin kanssa, jolloin Sony päätti viedä kehityksen loppuun ja julkaista oman konsolinsa. (Collins, K. 2)

Nintendo 64 (kuva 14.) julkaistiin vuonna 1996. Nintendo 64 oli viimeinen pelikasetteja käyttävä kotikonsoli. Pelikasettien suurin ongelma on alusta asti ollut niiden tallennuskapasiteetti. Varsinkin aikana jolloin Sonyn pelit sisälsivät hienoja esirenderöityjä videoita, pelikasettien heikkoudet korostuivat. CD-levyjen etuna oli myös pelien julkaisemisen helppous, siinä missä uusien pelikasettien valmistaminen oli hidasta, CD-levyjä voitiin kopioida useita kymmeniätuhansia päivässä. Tämä mahdollisti sen, että pelejä voitiin ottaa pienempi erä aluksi ja painaa nopeasti lisää mikäli tarvetta lisäpainokselle ilmeni. Nintendon aggressiivinen lisensointipolitiikka ja piraattituotteiden vastainen toiminta esti heitä kuitenkin siirtymästä CD-formaattiin. (Collins, K. 2)



Kuva 14. Sega Saturn (Tiptonware.com), Playstation (community.plus.net) ja Nintendo 64 (LaChance, J.)

Sega Dreamcast (kuva 15.) ilmestyi ensimmäisenä viime sukupolven konsoleista, vuonna 1999. Dreamcast oli ensimmäinen konsoli, jossa oli sisään rakennettu modeemi. Se käytti peliformaattinaan CD- ja GD-levyjä. Segan konsolin elinikä jäi lyhyeksi vain kahden vuoden

elinkaarellaan, sillä vuonna 2001 Sega ilmoitti lopettavansa konsolin valmistuksen. (Collins, K. 2)

Playstation 2 (kuva 15.) ilmestyi vuonna 2000. Sony lisäsi pelikonsoliinsa olohuoneen viihde toisto-ominaisuuksia käyttämällä peliformaattinaan DVD-levyjä. Pelikonsolia markkinoitiin olohuoneen viihdekeskuksena, sekä pelikonsolina ja sitä on myyty maailmanlaajuisesti noin 145 miljoonaa kappaletta. (Collins, K. 2)

Microsoft huomasi, että pelikonsoleissa liikkuu rahaa ja sen ensimmäinen konsoli Xbox (kuva 15.) ilmestyi vuonna 2001. Xbox oli konsolina iso PC:tä muistuttava laite. Päivittämällä konsolin firmwaren siihen sai asennuttua Linux-käyttöjärjestelmän. Xbox käytti myös formaattinaan DVD-levyjä. Xbox teki kuitenkin jotain eri tavalla kuin Playstation 2 ja se oli sen tuki internetin kautta pelaamiselle Xbox Live -palvelun avulla. Xbox:in suurin myyntivaltti oli Halo, joka esitteli kuinka PC:eellä suosittu ensimmäisen persoonan räiskinnät voitiin toteuttaa hyvin käyttäen kapulaohjainta hiiren ja näppäimistön sijaan. Xboxia on myyty maailmanlaajuisesti noin 24 miljoonaa kappaletta. (Collins, K. 2)

Nintendo Gamecube (kuva 15.) ilmestyi vuonna 2001 viimeisenä viime sukupolven konsoleista. Konsoli käytti formaattinaan MiniDVD-levyjä suojatakseen sen paremmin piratisointia vastaan. Formaattin valinta esti kuitenkin sen, että sitä olisi voinut käyttää DVD- tai CD-soittimena. Nintendo tuki Xboxin ja Playstation 2:den tavoin surround-ääniä. Gamecube on Nintendon huonoiten myynyt konsoli tähän päivään mennessä vain noin 21,7 miljoonan myynnillään. (Collins, K. 2)



Kuva 15. Sega Dreamcast (Decker, S), Playstation 2 (Howstuffworks.com), Xbox (Gameparadise.fi) ja Gamecube (Wcityauctions.com)

Pelikonsoleiden kuudes sukupolvi toi muutoksia konsolipelien maailmaan. Sega luovutti konsolivalmistajan roolinsa ja Microsoft teki suuren ensiaskeleen Xbox-konsolin myötä. Kuudes sukupolvi oli myös siirtyminen internet-yhteydellä varustettuihin palveluihin ja moninpelaamiseen, mikä on videopeleissä edelleen suuri trendi.

2.2 Nykypäivä

Tämänhetkisen kotikonsolisukupolven edustajat ovat Sony Playstation 3, Microsoft Xbox360 ja Nintendo Wii. Niitä on myyty maailmanlaajuisesti arviolta yhteensä n. 160 miljoonaa kappaletta marraskuun alkuun 2010 mennessä. Eniten myynyt konsoli on Nintendo Wii n. 47 % myydyistä konsoleista, toisena Xbox360 n. 28 % ja viimeisen Playstation 3 n. 25 %. (vgcharts.com.)

Kannettavien konsolien markkinoilla kilpailee tällä hetkellä Sony PSP ja Nintendo DS, joista DS on kilpailijaansa selkeästi edelle myyneissä. DS-käsi-konsolia on myyty n. 136 miljoonaa kappaletta ja PSP:tä n. 62 miljoonaa. Siinä missä PSP (kuva 16.) keskittyy tuomaan PS2-tasoisia pelejä kannettavaan muotoon, on DS:n myyntivalttina (kuva 16.) kaksi näyttöä, joista toinen on kosketusnäyttö. (vgcharts.com.)



Kuva 16. Sony PSP ja Nintendo DS (HardwareSphere.com)

PC-pelejä ja -pelaamista on pidetty viime vuosina kuolevana ilmiönä ja erityisesti pelimedia on kohdellut sitä sellaisena. PC-pelaaminen on kuitenkin muuttanut muotoaan ja esimerkiksi selainpelit ja latauspalvelut ovat kasvattaneet suosiotaan. Mikäli suurten peliyhtiöiden liikevaihtoja tarkastellaan alustoittain, on PC:n liikevaihto useassa tapauksessa suurempi kuin Nintendo Wii:n. (Mazel, J.)



Kuva 17. Ipod touch ja Iphone (Connick, M.)

Mobiilipelimarkkinat ovat olleet lähivuosina kovassa nousussa. Erityisesti Applen iPhone- ja iPod-laitteet ovat kasvattaneet mobiilipelaamisen suosiota (kuva 17.). Applen Appstore-ohjelmistokauppa on toiminut edelläkävijänä mobiilipelien myynnissä. Nintendo yhtiönä on noteerannut tämän alustan suosion ja pitää Applea nykyään suurimpana kilpailijanaan. Applen kanssa suoraan kilpailevat myös Googlen Android-alustaa käyttävät puhelimet, sekä Nokian Meego-alustaa käyttävät puhelimet. (Paavilainen, Korhonen & Saarenpää.)

Xbox360 (kuva 18.) julkaistiin ensimmäisenä tämän sukupolven konsoleista 22. marraskuuta 2005. Microsoftin toinen konsoli on tehokas pelilaitte, jonka suorituskyky oli tullessaan todella lupaava. Se sisälsi DVD-aseman ja siihen myytiin myös ulkoisia HD DVD -asemia. HD DVD hävisi kuitenkin formaattisodan Blue Raylle ja HD DVD -elokuvia ei enää valmisteta. Microsoft pyrkii jatkamaan konsolinsa elinikää 2010-luvun lopussa julkaistavalla Kinect-liikkeentunnistusohjelmalla. (Kjetland & Bass)

Toisena julkaistu Wii (kuva 18.), 19. marraskuuta 2006, otti erilaisen lähestymistavan pelaamiseen ja ei pyrkinyt olemaan kilpailijoidensa tasoinen supertietokone. Nintendo Wii:n myyntivalttina oli liikkeentunnistusohjain Wii Motion, jonka avulla pelaamiseen yhdistyi liikunta. Tämä kannatti, sillä Nintendon konsolin valmistuskustannukset pysyivät alhaisena ja konsolin hinta oli pitkään alhaisin markkinoilla. Wii on tämän konsolisukupolven selkeä markkinajohtaja, mutta sen myynti on selvästi alkanut hidastua. Erityisesti Wii vetosi ihmisiin, jotka eivät vielä harrastaneet pelaamista. (Plunkett, L.)



Kuva 18. Wii, PS3 ja Xbox 360 (Shroff, S.)

Viimeisenä julkaistu Sony Playstation 3 (kuva 18.) pyrki olemaan olohuoneen viihdekeskus pelkän pelikonsolin sijaan. Sisään rakennettu Blue Ray -soitin ja tuki USB-laitteille olivat sen myyntivaltteja. Konsoli on myös tehokkain tämän sukupolven konsoleista, joskin sen prosessoriarkkitehtuurin on sanottu olevan vaikea ohjelmoida. Playstation 3 sisältää Xboxin tapaan tuen surround-äänentoistolle. (Sony)

2.3 Äänentoisto

Äänentoistolaitteet voidaan jakaa muutamaaan kategoriaan, jotka ovat mono-, stereo- ja surround-laitteet. Äänentoisto yhdestä kaiuttimesta on monoa, kahdesta yleensä stereota ja useammasta surround-äänentoistoa.

2.3.1 Mono

Mono-ääni on yksikanavainen ääni, joka on käytössä mm. AM-radioissa, puhelimissa ja matkapuhelimissa. Ääni toistuu samalla tavalla riippumatta kaiuttimien määrästä, eikä näin ollen luo minkäänlaista kuvaa suunnasta, joskin kuva etäisyydestä voidaan luoda. Etäisyyskuva voidaan luoda yksinkertaisemmillaan äänenvoimakkuutta säätämällä. Mono-ääntä käytetään äänitystyössä, sillä hyvin usein parhaat mikrofonit ovat mono-mikrofoneja ja todellista hyötyä äänittää stereona ei yleensä ole, sillä mono-äännet lopulta voidaan miksata stereo-ääniraidaksi. (Haines, R. 17)

2.3.2 Stereo

Stereo-ääni eli kaksikanavainen ääni on edelleen se äänimuoto, jossa suurin osa viihde äänentoistosta toimii. Stereo-ääni koostuu periaatteessa kahdesta mono-raidasta, jotka ovat oikea ja vasen, joita toistetaan eri kaiuttimista. Stereo-äänellä on mahdollista luoda monipuolinen kuva siitä, mistä ääni kuuluu. Vaikka todellisuudessa äännet kuuluvat edelleen oikealta tai vasemmalta, erilaisia efektejä käyttämällä voidaan kuuntelijalle luoda äänimaailma, josta havaitsee eron kuuluko ääni ylhäältä vai alhaalta, tai kuuluuko ääni edestä vai takaa. Suunnan ja etäisyyden toteutus stereo-kuvassa voi tapahtua vaikka seuraavasti. (Rumsey & McCormick.)

Esimerkki. Tilassa on yksi äänilähde, joka toistaa kokoajan puhetta. Äänilähde sijaitsee kuuntelijan nähden vasemmalla ja näin ollen puhe kuuluu vasemmasta kaiuttimesta. Äänilähde siirtyy jonkin matkaa oikealle, jolloin ääni kuuluu myös oikeasta kaiuttimesta hiljaa. Ääni siirtyy hivenen kauemmas, jolloin ääni toistetaan vähän hiljempaa kummassakin kaiuttimessa. (Rumsey & McCormick.)

Edellä kerrotun esimerkin ongelma on siinä, että se ei vielä anna kuuntelijalle tietoa siitä, onko äänilähde edessä vai takana, koska pelaaja kuulee kaikki äänet edestään tai sivuilta, riippuen käyttääkö kuulokkeita vai kaiuttimia, ja kuinka kaiuttimet ovat sijoiteltu. Jos äänilähde on edessä saman verran kuin takana, ääni kuulostaa samalta toistettuna stereona kaiuttimista. Tämän takia ääneen täytyy tehdä muutoksia, että kuuntelija huomaa eron yksitapa on esimerkiksi lisätä äänentoistokorkeutta (pitch), kun ääni on edessä ja vähentää sitä, kun ääni on takana. Tähän kun lisätään kuuntelijalle mahdollisuus nähdä eteenpäin, aivot alkavat yhdistää hieman korkeamman äänen edestä tulevaan ääneen ja matalamman äänen takaa tulevaan. Sama voidaan tehdä myös korkeuden osalta eli tuleeko ääni ylä- vai alapuolelta. Tässä pitää kuitenkin muistaa se, että käyttää eri efektiä eri suunnan havainnollistamiseksi, sillä mikäli efekti on sama, ei ihminen havaitse eroa. (Rumsey & McCormick.)

Stereo-äänentoistolla voidaan simuloida ihmisen kuuloa tarjoilemalla molemmille korville oma raitansa ja luomalla tehosteilla sijainnin tuntumaa äänimaailmaan. Huonosti toteutettuna stereo-äänentoisto tarvitsee tuekseen visualisoinnin, antaakseen selkeän kuvan siitä mistä ääni kuuluu. Vaikka oikean, vasemman ja niiden välillä kuuluvan äänen erottaa helposti, auttaa visualisointi edestä, takaa, ylhäältä tai alhaalta tulevien äänten tunnistamisessa. Visualisoinnin avulla voi kuulija hetken pelattuaan tai katseltuaan elokuvaa erottaa äänen sijainnin. (Rumsey & McCormick.)

Hyvin toteutettuna ääniin laitetaan oikeat efektit, jotka ottavat huomioon ihmisen realistisen tavan kuulla eli päänmuodon, sen kuinka äänet saapuvat eri korvaan eri aikaan, sekä korvanlehden aiheuttaman vaimennuksen. Hyvin toteutettunakin efekti vaatii yleensä kuulokkeet toistovälineiksi, koska kaiuttimien asettelu vaikuttaa äänen kantautumiseen ja ihmisen fyysiset ominaisuudet (kallonmuoto, korvanlehdet, jne.), joita tehosteilla pyritään simuloimaan tulevat kahdesti. (Maijala, P.)

2.3.3 Surround

Surround-äänentoisto pyrkii luomaan usealla kaiuttimella pienen äänimaailman kuuntelijan ympärille. Kun äänet kuuluvat pelissä tai elokuvassa kameran takaa, ne kuuluvat oikeasti pelaajan takaa kaiuttimista. Tämä äänimaailman keskellä oleminen luo huomattavasti realistisemman kuvan pelin tai elokuvan tapahtumista. Surround-toistolla voidaan myös välittää

sijainti kerralla usealle ihmiselle, koska kaiuttimet luovat äänimaailman kuuntelijoiden ympärille. (Rumsey & McCormick.)



Kuva 19. Vasemmalla stereo-, oikealla surround 4.0 -äänijärjestelmä

Surround-äänit ovat monikanava-ääniä, jossa kaiuttimia lisäämällä pyritään luomaan uskottavampi äänimaailma, kuin stereo-äänessä. Surround-äänentoistosta on olemassa useita eri versioita mm. 2.1, 4.0, 4.1, 5.1. Nämä numerot tarkoittavat kaiutinmäärää ja subwoofer-kaiuttimien (matalataajuusäänien toistoon tarkoitettu kaiutin) määrää. Esimerkiksi. 4.0 sisältää 4 kaiutinta ja ei yhtään subwooferia (kuva 19.), kun taas 4.1 sisältää 4 kaiutinta ja yhden subwooferin. (Rumsey & McCormick.)

2.4 Äänten toistaminen

Äänien toistaminen tietokoneella on periaatteessa samanlaista, kuin kaikilla muillakin laitteilla. Ensin ääniraita tai äänitiedosto pitää lukea tallennusmedialta (CD, DVD, kiintolevy jne.), niin kuin missä tahansa muussakin äänentoistomediassa. Sen jälkeen tiedosto puretaan luetusta muodosta toistettavaan muotoon, mitä esimerkiksi vinyylilevyillä ei tarvitse tehdä. Lopuksi tämä purettu tieto lähetetään toistovälineelle eli äänikortille, joka laittaa signaalin ulos kaiuttimille. Tämä tiedon purkamisen voidaan hoitaa myös äänikortilla, mikäli purettavan tiedoston tyyppi on tuettu laitteistotasolla. (Kyrnin, M.)

Tietokoneissa, kännyköissä ja pelikonsoleissa on tietty määrä resursseja, joita voidaan käyttää hyväksi, mikäli niillä halutaan suorittaa jotain ohjelmaa. Vanhoissa tietokoneissa nämä resurssit oli jaettu keskenään, kunnes tietokoneeseen kehitettiin uusia lisälaitteita, kuten äänikortti ja näytönohjain. Nämä lisälaitteet lisäävät tietokoneeseen resursseja, jotka on suunnattu tiettyä tarkoitusta varten. Näytönohjain hoitaa grafiikan esittämisessä tarvittavia laskutoi-

mituksia ja lataa omaan muistiinsa kuvia piirtämiseen tarvittavana raakadatana. Äänikortti taas hoitaa äänitiedoston käsittelyn ja sen laskutoimitukset, joita äänenkäsittelyssä tarvitaan, sekä pitää omassa muistissaan sen äänentoistoon vaadittavaa raakadataa. On integroituja eli yhdysrakenteisia äänikortteja, jotka käyttävät tietokoneen yleisiä resursseja hyödykseen. Tietokoneen yleisiä resursseja käyttävät ohjelmistokiihdytetyt ratkaisut. Jos jokin äänikirjasto toimii ohjelmistokiihdytettynä, se käyttää tietokoneen prosessoritehoa ja muistia hyödykseen. (Kyrnin, M.)

Äänenlaatu on ratkaiseva tekijä siinä kuinka paljon resursseja käytetään. Mitä korkea laatuisempi äänitiedosto on, sitä enemmän sen esittäminen PCM-muodossa vaatii kiintolevytilaa. Mitä suurempi tiedosto pitää toistaa, sitä suurempi on sen vaatima muistin määrä. Pakatun tiedon purkaminen taas vaatii prosessoritehoa. (Kyrnin, M.)

Streamaus-tekniikka on kaikille tietokonetta käyttäville jollain tavalla tuttu. Tätä tekniikkaa käytetään muun muassa nettiradioissa, YouTube-palvelussa, sekä eri mediasoittimissa. Streamaus-tekniikan ideana on se, että tiedostoa voidaan lukea pieni osa kerrallaan, kun se on otettu vastaan. Tämä vastaanotettu palanen puretaan ja laitetaan puskuriin eli välimuistiin, josta se sitten toistetaan. Näitä puskureita on useita, joihin tiedostosta ladataan tietoa sitä mukaa, kun tiedostoa toistetaan. Puskureita vaihdellaan niin nopeasti, ettei taukoja pääse syntymään, joten käyttäjä kokee videon tai äänen yhtenäisenä. (Haines, R. Chapter 11)

Streamausta on suositeltavaa käyttää silloin, kun kyseessä on isokokoinen tiedosto, jonka lataaminen välimuistiin varaa sitä kohtuuttomasti ja sitä toistetaan pitkään, kuten esimerkiksi pelikentän ambient-raitaa tai taustamusiikkia. Streamaus käyttää resursseista eniten prosessointitehoa, koska joutuu purkamaan tiedostoa koko ajan. Yksi tai kaksi tällaista prosessia ei vielä vie resursseja liikaa.

Streamaus-tekniikkaa ei kuitenkaan kannata käyttää aina. Esimerkiksi on sotapelejä, jossa kuuluu taustamusiikki, sodan äänet, useita pelaajan ja NPC-hahmojen aiheuttamia ääniä, kuten juoksuaskeleet ja laukaukset yhtä aikaa. Tämä vaatii suuren määrän yhtäaikaista tiedoston latausta kiintolevyiltä, sen purkamista ja puskureiden tyhjentämistä ja uudelleen täyttämistä, mikä ei ole resurssien käytön kannalta tehokasta. Tällaisessa tapauksessa on tehokkaampaa, mikäli taustamusiikki ja ambient-raita eli kaukaiset taistelunäät soitetaan streamaus-tekniikalla, mutta laukaukset ja juoksuaskeleet pidetään muistissa purettuna pelin suorituksen ajan, jolloin prosessorin ei tarvitse hoitaa purkamista kaikille tiedostoille kerralla, vaan ne voidaan ladata muistiin etukäteen esimerkiksi pelitason latausvaiheessa.

3 ALUSTAT

Kajak3D-pelimoottori tukee tällä hetkellä Windows-, Maemo- ja iPhone-alustoja, sekä siihen kehitetään parhaillaan tuki myös Android-alustalle. Tämän opinnäytetyön lopputuote eli Kajak3D-äänirajapinta tukee samoja alustoja, joskin Android-alustan tukeminen jää jatkokehityksestä vastaavien tahojen käsiin. Tässä kappaleessa käsitellään näitä alustoja lyhyesti.

3.1 Windows

Windows on maailman suosituin käyttöjärjestelmä ja sen uusin versio Windows 7 ilmestyi pari vuotta sitten. Windows alustalla ohjelmointityökaluna käytetään yleensä Microsoft Visual Studiota. Kajak3D:n pääasiallinen kehitys on sidottu Visual Studio-ympäristöön.

Visual Studiolla kaikille kirjastoille löytyi valmiit käännösprojektit ja niiden kääntämisessä ei ollut mitään ongelmia. Niin dynaamisesti linkitettyinä, kuin staattisesti linkitettyinä kirjastot toimivat suoraan toimitettujen solution-tiedostojen avulla.

3.2 iPhone

iPhone OS -alustalle ohjelmointi tapahtuu Mac OS -käyttöjärjestelmällä ja sen kehitystyökalu on XCode. XCode-ohjelmointityökalun kääntäjä on hieman tarkempi kuin Windowsin Visual Studio ja antaa virheilmoituksia, myös koodin laatuun liittyvistä rikkeistä. XCode-työkaluun on integroitu iPhone-kehityksessä käytettävät SDK:t, Debug-työkalut ja lisenssi-palvelut.

iPhone OS ja XCode tukevat natiivina OpenAL-äänikirjastoa, joten sitä ei tarvitse kääntää Iphonelle erikseen vaan sen SDK tarjoilee valmiin frameworkin sen käyttämiseen. Framework on ohjelmistopaketti, joka sisältää kaikki kyseisen ohjelmakirjaston tarvitsemat tiedostot.

3.3 Maemo

Maemo on Nokian puhelimissa käytettävä Linux-pohjainen alusta. Maemo-alustalle ohjelmoitaessa käytetään Linux-käyttöjärjestelmää ja sen kehitystyökalu Maemo SDK on saatavissa vain Linuxille. Kajak3D:n kehityksessä Linux-käyttöjärjestelmää käytetään VMware virtuaalikoneen kautta ja virtuaali-levykuvan avulla. Käytännössä tämä tarkoittaa sitä, että ohjelmointi ja käännöstiedostojen kirjoittaminen voi tapahtua Windowsissa ja virtuaalikoneen avulla lähdekoodit ja käännöstiedostot käännetään suoritettaviksi tiedostoiksi. Käännöstiedostoja kutsutaan make-tiedostoiksi. Kääntäminen tapahtuu kutsumalla kääntökomentoa make siinä kansiossa missä käännöstiedosto sijaitsee. Käännöksen jälkeen suoritettavat tiedostot siirretetään laitteeseen, jossa ne halutaan testata tai ajaa.

4 ÄÄNIKIRJASTON TOTEUTUS

Tämä kappale kertoo äänirajapinnan toteutuksen vaiheista, sekä toimii samalla eräänlaisena käyttöoppaana kehitystä varten.

4.1 Vaatimusmäärittely ja suunnittelu

Äänirajapinnan toteutus alkoi asiakasvaatimuksista, jotka pelilaboratorio esitti. Näistä asiakasvaatimuksista tein vaatimusmäärittelyn, jonka pelilaboratorio hyväksyi ja jonka pohjalta äänirajapinnan toteutus alkoi. Liitteessä yksi esitetty vaatimusmäärittely toimi pohjana koko ohjelman toteutukselle. Vaatimusmäärittelyn pohjalta ulospäin näkyvä rajapinta suunniteltiin ja toteutettiin.

Vaatimusmäärittelyn jälkeen tutkin, mikä äänikirjasto sopisi parhaiten pelilaboratorion tarpeisiin. OpenAL vaikutti hyvältä vaihtoehdolta, koska sitä käytetään sekä Windows-, että Iphone-alustoilla laitteistokiihdytettynä ja koska OpenAL oli minulle ennestään tuttu. Maemoalustalle löytyi pienen etsinnän jälkeen ohjelmistokiihdytetty versio OpenAL-kirjastosta. OpenAL-kirjaston etuna on se, että se on avoin, ilmainen ja se suorittaa 3d-tehosteiden lisäämisen valmiiksi. Koska OpenAL oli selkeästi paras valinta ainakin kahdelle alustalle, aloin toteuttaa perustoiminnallisuutta.

4.2 Perustoteutus

Aloin tehdä perustoteutusta Windowsilla, koska sen kehitystyökalut ovat minulle tuttuja ja kirjastot helpoin saada toimimaan sillä. Aluksi käytin laitteiden alustamiseen ALUT-kirjaston alustus-funktioita. Lisenssistä johtuen kuitenkin loppuvaiheessa korvasin ALUTin käyttämällä OpenAL:n alustus funktioita.

Seuraavaksi toteutin Ogg-tuen ja tiedostojen lukemisen libvorbisfile-, libogg- ja libvorbis-kirjastojen funktioita käyttäen. Tiedostonluku tapahtui siis kokonaan libvorbisfile-kirjaston avulla. Kun ogg- ja wav-tuki oli valmis, tein pienen testiohjelman, jossa soi kaksi ääntä. Toi-

nen ääni asetettiin soimaan voimakkaammin vasemmalla puolella ja toinen ääni voimakkaammin oikealla. Ohjelma käytti toisena äänenä wav- ja toisen ogg-tiedostoa.

4.3 Maemo-käännös

Tämän jälkeen siirryin tekemään käännöstä Maemo-alustalle. Maemo-toteutuksessa suurimmat ongelmat tulivat VMwaren ja Maemo SDK:n tuntemuksen puutteesta. Kirjastojen kääntäminen on huomattavasti erilaista Windowsiin verrattuna ja pientä säätämistä löytyi useasta paikasta.

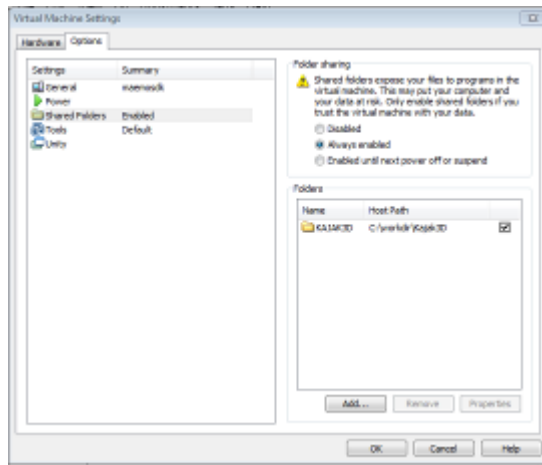
4.3.1 Maemo-asetukset

Maemo-alusta kehitys tapahtui siis VMware-virtuaalikoneella (kuva 20.). Kun MaemoSDK otettiin käyttöön, VMware-asetuksia täytyi muuttaa. Asetus-välilehti löytyi VM valikon kautta.



Kuva 20. VMware-virtuaalikone

Tärkein asetus joka täytyi laittaa kohdalleen, oli Shared Folder – asetus (kuva 21.). Asetuksen avulla Windows-kansio saadaan näkymään virtuaalikoneessa omana hakemistonaan. Tässä tapauksessa c:\workdir\kajak3d polkuun, joka on oletuspolku pelilaboratorion koneilla. Kun asetukset on saatu kohdalleen, on Maemo SDK valmis työskentelyyn.



Kuva 21. VMware Shared Folders -asetus

4.3.2 Scratchbox-asetukset

Maemon käännoistyökaluna toimiva Scratchbox tarvitsee erilliset asetukset. Scratchbox on tavallaan virtuaalikone virtuaalikoneen sisällä. Aikaisemmin Maemo SDK:hon lisätty kansion täytyi saada näkymään Scratchboxissa. Sitä varten piti vielä suorittaa seuraavat toimenpiteet.

Antaa komento ”sudo gedit /scratchbox/sbin/sbox_mount &”, joka avaa sbox_mount-tiedoston Gedit ohjelmassa ja lisätä sinne rivi ”do_mount --bind /mnt/hgfs/<KAJAK3D> \$ROOT/my_mount” Tämän jälkeen Windows-hakemisto löytyi Scratchboxissa my_mount/-polusta.

Scratchboxiin kirjaututaan sisään komennolla ”scratchbox/login”. Sen käännoisalusta muutetaan antamalla komento ”sb-conf se FREMANTLE_X86” mikäli ohjelmaa suoritetaan pc:llä ja ”sb-conf se FREMANTLE_ARMEL” mikäli se halutaan suorittaa puhelimessa.

4.3.3 Kääntäminen

Ensimmäinen ongelma oli kirjastojen OpenALsoft-, libvorbisfile-, libogg- ja libvorbis-kirjastojen kääntäminen Maemolle. Osa kirjastojen käännoistiedoista vaati Cmake-kääntäjän asentamisen. Cmake asennetaan kutsumalla ”fakeroot apt-get install cmake” komentoa Scratchbox komentokehotteessa.

Cmaken asennuksen jälkeen ulkoiset kirjastot kääntyivät ongelmitta. Osa kirjastoista vaati kuitenkin eri käännöslippujen käyttämisen toimiakseen staattisina. Kirjastot haluttiin staattisena, ettei tarvitsisi erikseen huolehtia eri kirjastojen toimittamisesta käyttäjälle.

Ulkoisten kirjastojen kääntämisen jälkeen aloin kääntää äänirajapintaa. Äänirajapinnan Maemo-käännökselle kirjoitin Make-tiedoston (Liite 2). Make-tiedosto suoritetaan komentokohotteessa ja se kääntää kirjaston. Mitään muutoksia äänirajapinnan lähdekoodiin ei vaadittu.

```
#if defined(IPHONE)
#include <OpenAL/al.h>
#include <OpenAL/alc.h>
#elif defined(MAEMO_CHANGES)
#include <openal-soft-1.12.854/include/AL/al.h>
#include <openal-soft-1.12.854/include/AL/alc.h>
#include <openal-soft-1.12.854/include/AL/alut.h>
#else
#include <AL/al.h>
#include <AL/alc.h>
#include <AL/alut.h>
#endif
```

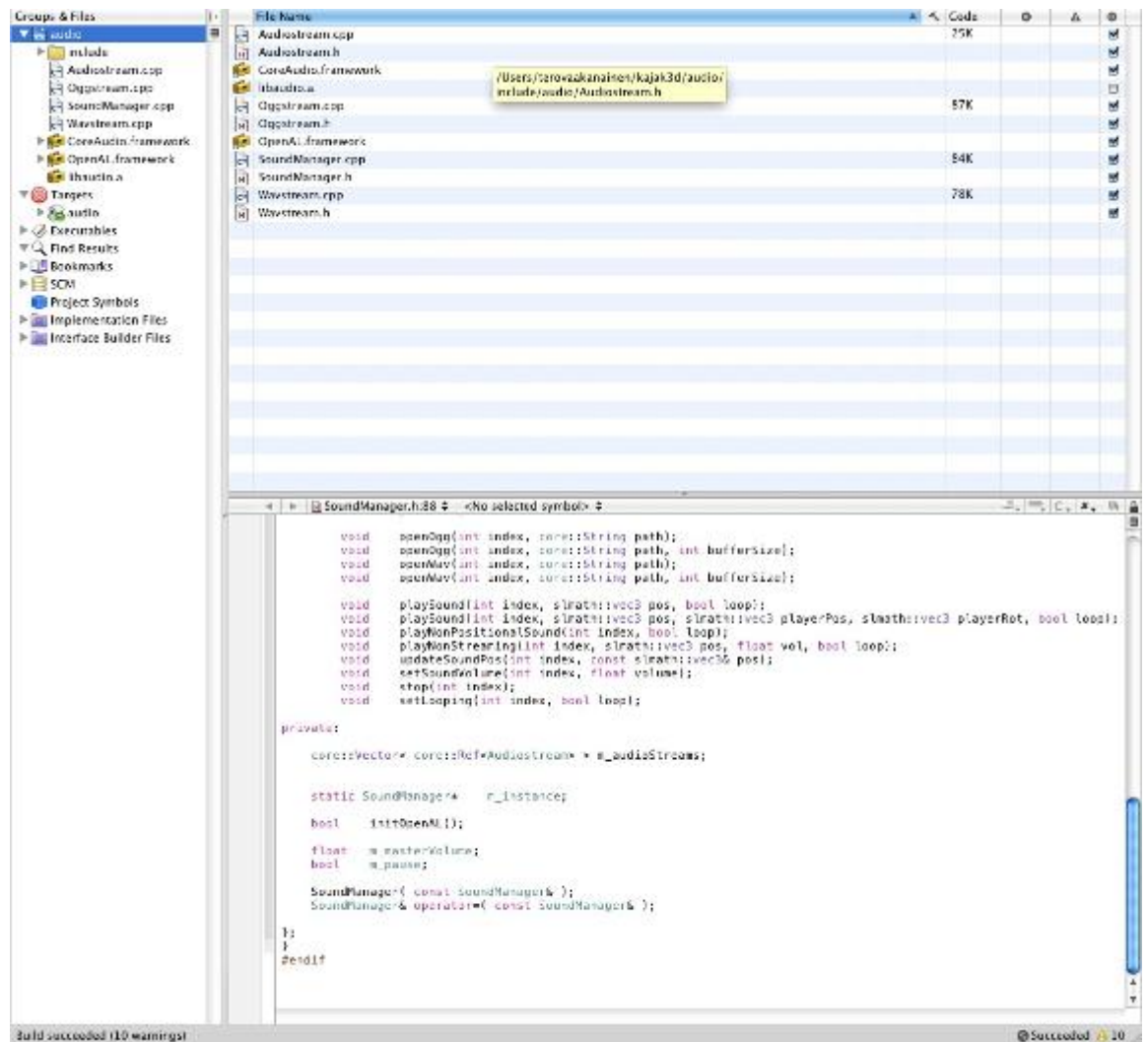
Kuva 23. Audiostream.h-tiedoston alustamäärittelyt

Ainoat alustakohtaiset määrittelyt esiintyvät Audiostream.h-tiedostossa (kuva 23.), joka on pohjaluokka, josta eri tiedostomuodoille tarkoitetut luokat periytetään. Luokan `#include`-polut osoittavat siis eri tiedostoihin.

Testiohjelman kääntäminen Maemolle vaati oman make-tiedoston (Liite 3) ja sen käännöksessä tulevat esiin vasta oikeat linkittäjävirheet. Kun ohjelma oli käännetty onnistuneesti, siirsin sen Nokian N900-puhelimeen ja testasin, että ohjelma toimii siinä.

4.4 Iphone käännös

Maemo-käännöksen valmistuttua oli vuorossa käännös Iphonelle. Iphone käännöksen tekeminen oli huomattavasti helpompaa, kuin Maemo-käännöksen, sillä OpenAL-kirjasto on valmiina Iphone SDK:ssa. XCode-ohjelmointityökalu on hyvin samankaltainen Visual Studio kanssa, joten työkalua oli helppo käyttää.



Kuva 24. XCode:n perusnäkö

Xcodella (kuva 24.) täytyi luoda uusi projekti, lisätä lähdekoodit, lisätä include-polut ja kirjastopolut, sekä tehdä audiolib-käännöskohde eli target. Sen jälkeen audiokirjastoa oli samanlaista kehittää kuin Visual Studiolla.

Testiohjelman kääntämistä varten pelilaboratorio toimitti minulle valmiin Iphone-alustus esimerkin. Korvaamalla esimerkin alustariippumattomat ohjelmakoodit, oli testiohjelman käännös valmis. Ensin käänsin ohjelman simulaattorilla toimivaksi. Simulaattori ei vaadi lisenssiä toimiakseen. Sen jälkeen kun testiohjelma toimi, oli aika testata sitä puhelimella. Puhelinkäännös vaatii kuitenkin Iphone developer -lisenssin ja -sertifikaatin, että ohjelma voidaan kääntää puhelimelle. Lisenssiä varten Applen Developer Center -sivustolla on oma alisivustonsa ja sieltä minun täytyi anoa lisenssiä. Lisenssianomuksen hyväksyi sen jälkeen Mikko Romppainen, joka vastaa pelilaboratorion Iphone kehityksestä. Lisenssihyväksynnässä asetetaan myös laitteet, joilla lisenssiä voi käyttää. Tämän jälkeen lisenssit il-

mestyvät ladattavaksi henkilökohtaiselle Developer Center -sivulle, josta ne tallennetaan koneella ja annetaan XCodelle.

Kun lisenssit oli asennettu, kytkettiin puhelin tietokoneeseen ja testiohjelma voitiin kääntää sille. Testiohjelma lähtee automaattisesti käyntiin puhelimessa, jos USB-johto on kiinnitetty siihen ja XCodella suoritetaan RUN-komento. Testiohjelma toimi ja näin perustoteutus oli valmis.

4.5 Testaus

Kun äänirajapinta oli käännetty kaikille alustoille ja projekti-tiedostot valmiina alkoi todellinen testaus, jossa ohjelman kaikki ominaisuudet käytiin läpi. Ohjelmiston jokainen funktio piti testata ja katsoa, että ominaisuudet toimivat jokaisella alustalla. Testiohjelmassa käytetään molempia äänitiedostomuotoja sekä streamattuna, että perustoistolla. Testiohjelmassa soi parhaimmillaan neljä yhtäaikaista ääntä.

Testiohjelma toimii moitteetta Windows-alustalla ja nykytietokoneet jaksavat pyörittää helposti neljää yhtäaikaista ääntä, niin streamattuna, kuin perustoistolla. Näin vähäisten äänten käyttö ei vaikuta pelinopeuteen tietokoneella.

Maemo-alustalla huomasin, että äänet pätkivät helposti, mikäli niitä streamataan. Tämä johtuu todennäköisesti siitä, että streamauksessa tietoa joudutaan lukemaan kokoajan ja laitteen tehot eivät meinaa riittää kunnolla, koska äänet toimivat ohjelmistokiihdytettynä.

IPhone-alustalla äänet toimivat moitteetta ja rautakiihdytyksestä johtuen eivät äänet vaikuta suoritusnopeuteen kovin paljon. Neljä yhtäaikaista ääntä toimi hyvin, eikä Maemolla esiintyvää pätkintää ilmennyt.

5 KAJAK3D-ÄÄNIRAJAPINNAN RAKENNE

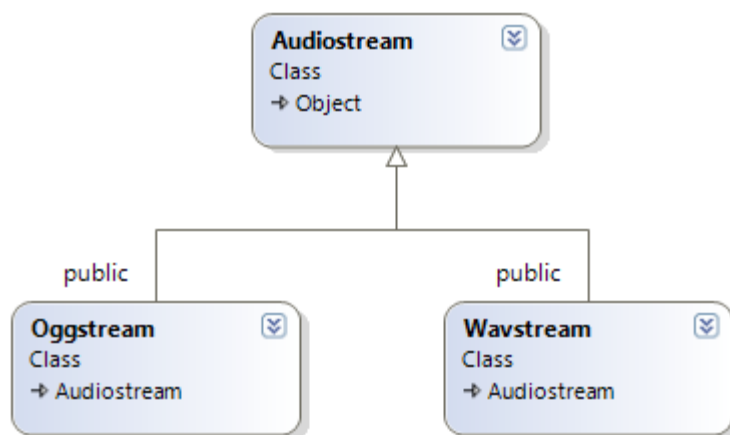
Opinnäytetyönä toteutetun äänirajapinnan sisäisten toimintojen esittely ja toimintaperiaatteet esitellään tulevissa kappaleissa. Tässä kappaleessa käydään läpi, myös ulkoiset kirjastot joita ääntenkäsittelyrajapinta käyttää äänentoistossa.

5.1 Äänirajapinnan rakenne

Kajak3D-äänirajapinta koostuu neljästä luokasta jotka ovat Soundmanager, Audiostream, Oggstream ja Wavstream. Soundmanager on käyttöliittymäluokka, Audiostream pohjaluokka, ogg- ja wavstream-luokat taas tiedostomuotoisia luokkia.

5.1.1 Soundmanager

Soundmanager-luokka on käyttäjälle näkyvä luokka, joka hoitaa äänirajapinnan toiminnallisuuden ylimmänkerroksen. Soundmanager sisältää myös OpenAL-alustuksen. Soundmanager on singleton-luokka.



Kuva 25. Audiostream-kantaluokasta periytetään Oggstream- ja Wavstream-luokat

5.1.2 Audiostream

Audiostream-luokka on puhtaasti virtuaalinen pohjaluokka, josta eri äänitiedostomuotoja tukevat luokat, tässä tapauksessa Oggstream ja Wavstream, periytetään. Audiostream ei sisällä toiminnallisuutta.

5.1.3 Oggstream

Oggstream-luokka on ogg-tiedostojen käsittelyyn tarkoitettu luokka, joka hoitaa äänirajapinnan kommunikoinnin libogg-, libvorbis- ja libvorbisfile-kirjastojen välillä. Oggstream-luokassa annetaan libvorbisfile-kirjastolle metodit tiedostonkäsittelyyn, jotka korvaavat sen valmiin tiedostonkäsittelyn.

5.1.4 Wavstream

Wavstream-luokan toiminnallisuus on lähes identtinen Oggstream-luokan kanssa, joskin tiedostojen käsittelyyn liittyvät funktiot ovat erilaisia. Tiedostojen käsittely hoidetaan, suoraan Kajak3D:n funktioilla.

5.2 Äänilaitteen alustus

Äänilaitteen alustus OPENAL-kirjastolla voidaan suorittaa monella tapaa. Siihen voi käyttää ulkoista apukirjastoa, kuten ALUT, joka hoitaa virhetapausten käsittelyn, sekä alustukset yksinkertaisesti kutsumalla `Init()` funktioita tai tekemällä sen itse käyttäen apuna OPENAL-ohjeita. ALUT-kirjastoa käytettäessä, on etuna se, että tavat, joilla laite alustetaan, ovat testattuja ja virhetilanteet ilmoitetaan selkeästi. ALUT-kirjastoa ei kuitenkaan voi käyttää suoraan sellaisenaan iPhone-alustalla. Tästä syystä audiokirjasto ei käytä ALUT-kirjastoa äänenalustuksessa.

Äänirajapinta otetaan käyttöön luomalla `soundmanager`-olio. `Soundmanager`-olioita voidaan luoda vain yksi, koska se on singleton-luokka. `Soundmanager`in luonti voi tapahtua seuraavalla komennolla:

```

Soundmanager sm;

|bool SoundManager::initOpenAL()
{
    m_default_device = alcGetString(NULL, ALC_DEFAULT_DEVICE_SPECIFIER);

    if((m_device = alcOpenDevice(NULL)) == NULL)
    {
        return false;
    }

    m_context = alcCreateContext(m_device, NULL);

    alcMakeContextCurrent(m_context);
    alGetError();

    return true;
}

```

Kuva 26. Audiokirjaston OpenAL-alustusfunktio

Muuttujan luonnin jälkeen äänirajapinta pitää initialisoida. Initialisointi tapahtuu kutsumalla Soundmanager-kirjaston `Init(float volume)`-funktioita. `Init(float volume)`-funktioita voidaan kutsua vasta Kajak3D:n alustuksen jälkeen. Esimerkiksi `createDevice()`-funktioissa. `Init(float volume)`-funktio kutsuu `InitOpenAL()`-funktioita (kuva 26.), sekä asettaa äänen voimakkuuden jonka `Init(float volume)`-funktio ottaa parametrina. Alustus tapahtuu seuraavalla kutsulla:

```
sm.Init(1.0f);
```

Sulkujen sisällä oleva `1.0f` on liukuluku arvo ja tarkoittaa, että äänen voimakkuus asetetaan täydelle voimakkuudelle. Arvo tallennetaan `m_masterVolume`-muuttujaan. Kun initialisointi on suoritettu, voidaan ladata äänitiedostoja.

5.3 Tiedoston luku

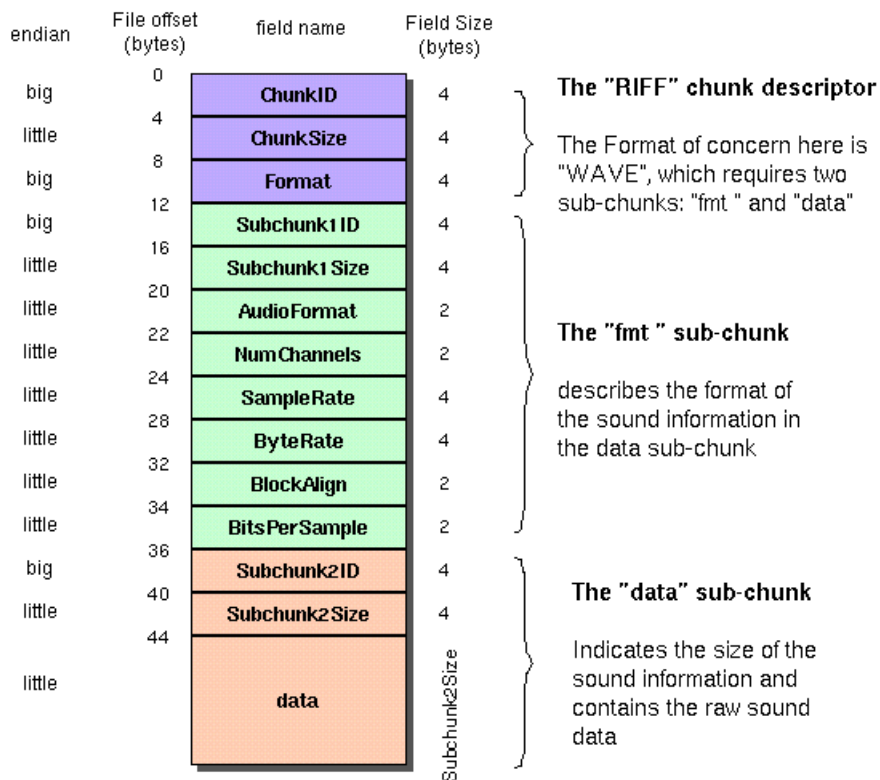
Tiedostojen lukemiseen käytettävät funktiot korvattiin `kajak3d`:n tiedostonkäsittelyfunktioilla, perinteisten tiedostonluku-funktioiden sijaan. WAV-tiedostojen lukeminen on paljon yksinkertaisempaa, kuin OGG-tiedostojen, sillä WAV-tiedosto on pakkaamaton ja OPENAL käyttää PCM-ääntä suoraan. OGG-tiedostomuoto on taas rakennettu streamausta ajatellen ja se on pakattu, se täytyy ensin lukea ja purkaa PCM-muotoon, ennen kuin se välitetään OPENAL-kirjastolle toistettavaksi.

Tiedostojen avaamiseen äänirajapinta tarjoaa seuraavat funktiot:

- `sm.OpenOgg(int index, string path, int buffersize)`
- `sm.OpenOgg(int index, string path)`
- `sm.OpenWav(int index, string path, int buffersize)`
- `sm.OpenWav(int index, string path)`

5.3.1 Wav

Wav-tiedosto koostuu otsaketiedoista (header) ja itse äänitiedosta (data). Ensimmäiset 44 bittiä tiedoston alusta on header-tietoa. Wav-tiedoston header-osio pitää sisällä kuvan 27. mukaiset tiedot, kuvion osoittamassa bittijärjestyksessä.



Kuvio 27. Wav-tiedoston rakenne (Wilson, S)

Tämän jälkeen tulee itse audiodata, joka voidaan välittää sellaisenaan tai pienemmissä osissa OpenAL-äänikirjastolle. Äänikirjastolle välitetään tiedostosta myös ByteRate-, NumChannels- ja BitsPerSample-tiedot, jotta OpenAL osaa käsitellä sille välitettävää äänitietoa oikealla tavalla.

5.3.2 Ogg

OGG-tiedostojen lukemiseen käytetään, myös Kajak3D:n tiedostonkäsittelyfunktioita, joskin itse tiedostojen käsittely tapahtuu vorbisfile-kirjaston kautta. Vorbisfile-kirjasto pitää sisällään valmiit funktiot tiedostojen käsittelyyn, jotka korvataan tai ylikirjoitetaan käyttämään Kajak3D:n funktioita. Funktioita on neljä kappaletta:

- Read_func
- Seek_func
- Close_func
- Tell_func

Kun nämä funktiot ovat korvattu, tapahtuu tiedostonkäsittely vorbisfile-kirjaston normaalifunktioiden avulla. Käyttäjälle tällä tiedolla ei kuitenkaan ole väliä, sillä äänirajapinta hoitaa tämän käyttäjän puolesta.

5.4 Äänitiedoston toistaminen

OpenAL-äänikirjasto käyttää toistettaville äänille seuraavia tietoja: Äänilähteen paikka, kuulijain paikka ja pään asento, toistokorkeus, äänenvoimakkuus ja toistetaanko ääntä silmukassa vai ei. Äänitiedoston toistamiseen Kajak3D-äänirajapinta tarjoaa seuraavat funktiot käyttötarkoituksen mukaan.

```
sm.playSound(int index, slmath::vec3 soundsourcepos,
slmath::vec3 listenerpos,
slmath::vec3 listenerRotation, bool loop);
sm.playNonPositionalSound(int index, bool loop);
sm.playNonStreaming(int index, slmath::vec3 pos, float volume, bool loop);
```

Toistofunktiot ottavat sisäänsä indeksinumeron, jossa toistettava ääni sijaitsee, paikan 3D-avaruudessa slmath::in, äänenvoimakkuuden float-liukulukuna ja boolean arvon siitä toistetaanko tiedostoa silmukassa vai ei.

5.4.1 Mono- ja stereo-raidat

Kun äänitiedosto toistetaan OpenAL-äänikirjaston kautta, vain mono-äänelle voidaan antaa sijaintitietoja. Stereoraitoja ei voida käyttää ääniefekteinä, mikäli OpenAL:n sijaintifunktioita halutaan käyttää. Stereoraidat sopivat kuitenkin hyvin taustamusiikin toistamiseen. Myös toistoformaattia valitessa kannattaa ottaa huomioon alusta. Mikäli alusta ei tue 24-bittisiä stereoääniä, ei ole järkevää tallettaa tietoa kyseiselle alustalle sellaisessa muodossa.

5.4.2 Etäisyys ja efektit

Äänirajapinta ottaa äänten koordinaatit 3d-avaruudessa slmath-kirjaston `vec3`-tyyppisinä muuttujina ja antaa ne float-arvoina OpenAL äänikirjastolle. OpenAL hoitaa näillä tiedoilla äänen käsittelyn ja miksaa lopputuotoksen stereoraidaksi.

5.4.3 Streamaus ja kokotiedostontoisto

Streamaus varten OpenAL tarjoaa jokaista lähde-muuttujaa varten puskuripinon, johon puskureita voidaan kasata suorittamista varten. Puskuri lisätään pinoon kutsumalla seuraavaa funktioita:

```
alSourceQueueBuffers(m_source, 1, &buffer);
```

Puskuri on ennen tätä täytetty äänitiedolla. Kun puskuri on toistettu, se poistetaan pinosta. Puskurin poistaminen pinosta tapahtuu kutsumalla seuraavaa funktioita:

```
alSourceUnqueueBuffers(m_source, 1, &buffer);
```

Kokotiedostontoistossa taas luodaan puretun äänitiedon kokoinen puskuri, joka täytetään ladattavan äänitiedoston äänitiedolla. Tämän jälkeen kyseinen puskuri annetaan OpenAL tiedostolle toistettavaksi.

5.5 Äänien päivitys ja hallinta

Äänirajapinnan tiedot päivitetään kutsumalla `sm.update()` funktiota. Tämä funktio hoitaa äänitiedostojen tilan päivittämisen ja soundmanager tarvitsee sitä toimiakseen, sillä streamatut tiedostot päivitetään sen avulla.

Äänitiedostojen hallintaan kirjastossa on seuraavat funktiot:

- `sm.pause()`
- `sm.resume()`
- `sm.stopAll()`
- `sm.stop(int index)`
- `sm.updatePlayerPos(const slmath::vec3& pos, const slmath::vec3& rot);`
- `sm.updateSoundPos(int index, const slmath::vec3& pos);`
- `sm.setVolume(float volume);`
- `sm.setSoundVolume(int index, float volume);`
- `sm.setLooping(int index, bool loop);`
- `sm.setPitch(int index, float pitch)`

6 YHTEENVETO

Tämän opinnäytetyön alussa käsiteltiin videopelien historiaa peliäänten kannalta, sillä se auttaa selvittämään mistä nykyiset tekniikat ja ratkaisut ovat lähtöisin, sekä auttaa mahdollisesti ymmärtämään mihin olemme menossa eri laitteiden äänentoistossa.

Seuraavaksi käsiteltiin eri äänentoistomuotoja, joita nykyään käytetään. Äänentoistomuotojen ymmärtäminen auttaa käsittämään, mitä eri äänitiedostomuodoilla pyritään saavuttamaan ja missä niitä käytetään.

Teoriaosasta viimeiseksi käsiteltiin tietokonelaitteissa tapahtuvan äänentoiston periaatteet, tiedonlukemisesta purkamiseen ja toistamiseen. Tämän tarkoituksena on antaa selkeämpi kuva siitä, kuinka laitteet toimivat sisäisesti ja mitä eri vaiheita erilaisten tiedostomuotojen toistamisessa tarvitaan.

Kappaleessa 3 käsiteltiin alustoja, joilla äänirajapinta toimii. Alustat eroavat toisistaan äänentoisto-ominaisuuksiltaan ja laitteistoltaan, sekä työkaluiltaan, joilla kehitys tapahtuu. Kappaleen tarkoituksena on kertoa, kuinka kehitys kullakin alustalla tapahtuu, kun käytetään Kakjak3D-pelimoottoria.

Kappaleessa 4 kerrottiin äänirajapinnan toteutuksen eri vaiheet, sekä annetaan ohjeita alustojen käyttämiseen. Kappaleen tarkoituksena on selvittää kuinka äänirajapinta toteutettiin. Neljännessä kappaleessa kerrottiin, kuinka äänirajapinta testattiin.

Viidennessä kappaleessa selvitettiin äänirajapinnan rakennetta eli millaisia luokkia rajapinta pitää sisällään, sekä paneudutaan siihen millainen rajapinnan toiminnallisuus on. Kappaleen tarkoituksena on se, että äänirajapinnan toiminnallisuudesta kiinnostuneet voivat käyttää sitä apunaan, kun kehittävät äänirajapinnan toiminnallisuutta paremmaksi tai kun kehittävät peliä rajapintaa käyttäen.

Äänirajapinnan kehitystä voi jatkaa eteenpäin usealla eri tavalla. Yksinkertaistamalla sen toiminnallisuutta, kääntämällä se uusille alustoille ja lisäämällä tuen uusille äänitiedostomuodoille.

LÄHTEET

- Barton, M & Loguidice, B. 2008. A History of Gaming Platforms: Mattel Intellivision. Gamasutra.com.
http://www.gamasutra.com/view/feature/3653/a_history_of_gaming_platforms_.php 3.11.2010
- Bowen, K. 2010. The Gamespy Hall of Fame: Space Invaders. Gamespy industries.
<http://archive.gamespy.com/legacy/halloffame/spaceinvaders.shtm> 5.11.2010
- Collins, K. 2005. From Bits to Hits: Video Games Music Changes its Tune. Film International #12, Tammikuu 2005. <http://www.gamessound.com/texts/bits2hits.pdf>
 19.9.2010
- Graft, K. 2006. Analysis: History of Cell-Phone Gaming. Next Generation.
http://www.businessweek.com/innovate/content/jan2006/id20060122_077129.htm . 3.11.2010.
- Guerrieri, M. 2009. Fate On the Line: Tracking the Beethoven's Fifth Ringtone.
<http://thefastertimes.com/classicalmusic/2009/09/05/fate-on-the-line-tracking-the-beethovens-fifth-ringtone/> . 4.11.2010
- Haines, R. 2001. Digital Audio. The Coriolis Group LLC.
- Juutilainen, M. 2010. Siirtyvä tietoliikenne: Matkapuhelinverkot. Lappeenrannan teknillinen yliopisto. <http://www2.it.lut.fi/kurssit/06-07/Ti5312600/luentokalvot/luento07-08.pdf> . 4.11.2010
- Kjetland & Bass. 2010. Microsoft Will Offer Xbox 360 for Five More Years. Bloomberg L.P. <http://www.businessweek.com/news/2010-08-23/microsoft-will-offer-xbox-360-for-five-more-years.html> . 5.11.2010
- Kyrnin, M. 2010. Understating Computer Audio. About.Com.
<http://compreviews.about.com/cs/soundcards/a/CompAudioPt1.htm> . 4.11.2010

- Maijala, P. Binauraalinen äänitys ja toisto kuuntelukokeita varten. 1996. Teknillinen Korkeakoulu. <http://www.acoustics.hut.fi/~pantse/projects/semmas96/binaural.html> . 24.6.2010
- Marks, A. 2009. Complete Guide to Game Audio 2nd edition. Focal Press an imprint of Elsevier Inc.
- Mazel, J. 2010. Activion-Blizzard Profits \$51m on Revenue of \$745m in July-Sep. VGCharts Ltd. <http://gamrfeed.vgchartz.com/story/82637/activision-blizzard-profits-51m-on-revenue-of-745m-in-july-sep/> . 4.11.2010
- McCormick & Rumsey. 2009. Sound and Recording 6th edition. Focal Press an imprint of Elsevier Inc.
- Melanson, D. 2006. A Brief History of Handheld Video Games. www.engadget.com. <http://www.engadget.com/2006/03/03/a-brief-history-of-handheld-video-games/>
- Messmer, E. 2008. Tech Talk: Where'd it Come From, Anyway?. http://www.pcworld.com/businesscenter/article/147698/tech_talk_whered_it_come_from_anyway.html . 25.9.2010
- OpenAL 1.1 Specification and Reference. 2005. Creative Labs Inc. <http://connect.creativelabs.com/openal/Documentation/OpenAL%201.1%20Specification.htm> . 17.4.2010
- Paavilainen, Korhonen & Saarenpää. 2009. Pelaaminen matkapuhelimella nyt ja tulevaisuudessa. Tampereen yliopisto. <http://research.nokia.com/files/ptvk2009-06.pdf> . 4.11.2010
- Plunkett, L. 2010. Wii Sales Down 21 % In A Single Year. Kotaku.com. <http://kotaku.com/5532274/wii-sales-down-21-in-a-single-year> . 4.11.2010
- Ratan, S. 2003. The Games N-Gage Doesn't Play. The Wired. <http://www.wired.com/gaming/gamingreviews/news/2003/11/61117> . 4.11.2010
- Samsung's History 1997-1999. 2010. Samsung.com. http://www.samsung.com/hk_en/aboutsamsung/corporateprofile/history01.html . 4.11.2010

- Schramm, M. 2010. Pachter: Apple will have a game console soon. AOL Inc.
<http://www.tuaw.com/2010/03/02/pachter-apple-will-have-a-game-console-soon/>
 . 4.11.2010
- Sony.fi. 2010. Playstation 3. Sony Europe Limited.
<http://www.sony.fi/section/haku/article/id/1223049864908> . 4.11.2010
- Vgcharts.com. 2010. VGChartz Ltd. <http://www.vgchartz.com/#Worldwide%20Totals> .
 4.11.2010
- Wilson, S. 2003. <https://ccrma.stanford.edu/courses/422/projects/WaveFormat/> .
 4.11.2010
- Xiph.org. Vorbisfile documentation. 2010. Xiph.org <http://xiph.org/vorbis/doc/vorbisfile/>
 . 4.11.2010
- KUVAT
- Academic.ru. http://de.academic.ru/pictures/dewiki/73/IBM_PC_5150.jpg
- Andrews, S. 2007. The 10 Most Important Games Ever - Space Invaders - Taito 1978.
 TrustedReviews & IPC Media. <http://www.trustedreviews.com/video-games/review/2007/11/28/The-10-Most-Important-Games-Ever/p2> . 4.11.2010
- Arcadeshop.com. 2010. ArcadeShop Amusements Lcc.
<http://www.arcadeshop.com/gamepic/new-namco-pac-man-ur-ls.jpg>
- Beeslife.com. http://www.beeslife.com/handheld/auto_race1.jpg
- Connick, M. http://mconnick.files.wordpress.com/2008/07/iphone_ipod_touch_01.jpg
- Crackerwax.com. <http://www.crackerwax.com/wp-content/uploads/2010/08/Commodore64.jpg>
- Decker, S. http://www.scottdecker.com/video_games/sega_dreamcast.jpg
- Gameparadise.fi. http://www.gameparadise.fi/catalog/images/111125-xbox_console.jpg
- Gamesetwatch.com. <http://www.gamesetwatch.com/gamegear.jpg>
- Gaminggenerations.com. <http://www.gaminggenerations.com/store/images/GBASP.jpg>
- Handheldmuseum.com. <http://www.handheldmuseum.com/MB/MB-MicrovisionUS.jpg>
- Hardwaresphere.com. <http://www.hardwaresphere.com/wp-content/uploads/2009/02/valentine-gifts-idea-for-boys-psp-ds.jpg>

- Hill, J. 2006. Pong: 30 years on. Screen Play Blog.
<http://blogs.theage.com.au/screenplay/archives//003882.html> . 5.11.2010
- LaChance, J. http://3.bp.blogspot.com/_WrynUIkVu-Y/TA9SBPDhErI/AAAAAAAAAFc/H6Tq2JL8KFE/s1600/Nintendo_64.jpg
- Lockley, C. <http://community.plus.net/wp-content/uploads/2009/03/playstation.jpeg>
- Mobile-Review.com. http://media.photobucket.com/image/n-gage%20mobile-review/al_oasis1/ngagephone.jpg
- Old-computers.com. http://www.old-computers.com/museum/photos/IBM_PCjr_System_1.jpg
- Pauljamesmannering.com
<http://www.pauljamesmannering.com/Assets/Consoles/Gameboy%20Colour.jpg>
- Pugo.org. http://www.pugo.org/media/collection/console/gameboy_advance.jpg
- Pugo.org. <http://www.pugo.org/media/collection/console/nintendo-entertainment-system.jpg>
- Ratteswg.de. http://www.ratteswg.de/Uber_mich/Hobbies/Nintendo/Gameboy.jpg
- Retrogamers.ch. <http://www.retrogamer.ch/bilder/smdmegadrive2.jpg>
- Rolentapress.com. <http://www.rolentapress.com/rolenta/collection/nintendo/gameboy-pocket.jpg>
- Scott, J. <http://ascii.textfiles.com/wp-content/uploads/2009/06/atari2600a.jpg>
- Shroff, S. <http://static.skattertech.com/media/2007/12/wii-ps3-xbox360.jpg>
- Sobcontrollers.com. http://www.sobcontrollers.com/temp/wp-content/uploads/2010/04/Game_and_watch_Ball.jpg
- Surrealist.de.
http://www.surrealist.de/spielmonster/Impressum/Gewinnspiel/Monsterletter/Ko nsolen-Ecke/SNES/Nintendo_SNES_02.jpg
- Swotti.com.
<http://www.swotti.com/tmp/swotti/cacheZ2FTZWJVESBTAWNYBW==/imgGameboy%20Micro2.jpg>
- Tiptonware.com. <http://tiptonware.com/SegaSaturn.jpg>
- Torumi.com. http://www.torumi.com/images_articles/18.jpg
- Tyson, J. <http://static.howstuffworks.com/gif/ps2-photo.jpg>
- Videogameconsolelibrary.com.
http://www.videogameconsolelibrary.com/images/1970s/72_Magnavox_Odyssey/72_Magnavox_Odyssey_General1.jpg

Wcityauctions.com.

http://www.wcityauctions.com/my_files/images/Game%20System%20Images/Game_Cube.jpg

Wikipedia.com. <http://upload.wikimedia.org/wikipedia/commons/9/9b/Atari-lynx-1-1000.jpeg>

Wired.com. http://www.wired.com/images/article/full/2008/07/pacman_500px.jpg

LIITTEET

- LIITE 1 Vaatimusmäärittely
- LIITE 2 Äänirajapinnan Make-tiedosto
- LIITE 3 Testiohjelman Make-tiedosto

1	äänirajapinta toimii Windows 7, Vista ja XP alustoilla
1.1	äänirajapinta voi toistaa WAV tiedostoja
1.1.1	äänirajapinta toistaa wav tiedoston
1.1.2	äänirajapinnalla voi pysäyttää wav tiedoston toiston
1.1.3	äänirajapinnalla voi tauottaa wav tiedoston toiston
1.1.4	äänirajapinnalla voi asettaa wav äänen voimakkuuden(volume)
1.1.5	äänirajapinnalla voi asettaa wav äänen toistokorkeuden(pitch)
1.2	äänirajapinta voi toistaa OGG tiedostoja
1.2.1	äänirajapinta toistaa ogg tiedoston
1.2.2	äänirajapinnalla voi pysäyttää ogg tiedoston toiston
1.2.3	äänirajapinnalla voi tauottaa ogg tiedoston toiston
1.2.4	äänirajapinnalla voi asettaa ogg äänen voimakkuuden(volume)
1.2.5	äänirajapinnalla voi asettaa ogg äänen toistokorkeuden(pitch)
1.4	äänirajapinta voi ladata WAV ja OGG tiedoston pakatusta hakemistosta
1.5	äänirajapinta voi ladata WAV ja OGG tiedoston hakemistosta
2	äänirajapinta toimii Iphone alustalla
2.1	äänirajapinta voi toistaa WAV tiedostoja
2.1.1	äänirajapinta toistaa wav tiedoston
2.1.2	äänirajapinnalla voi pysäyttää wav tiedoston toiston
2.1.3	äänirajapinnalla voi tauottaa wav tiedoston toiston
2.1.4	äänirajapinnalla voi asettaa wav äänen voimakkuuden(volume)
2.1.5	äänirajapinnalla voi asettaa wav äänen toistokorkeuden(pitch)
2.2	äänirajapinta voi toistaa OGG tiedostoja
2.2.1	äänirajapinta toistaa ogg tiedoston
2.2.2	äänirajapinnalla voi pysäyttää ogg tiedoston toiston
2.2.3	äänirajapinnalla voi tauottaa ogg tiedoston toiston
2.2.4	äänirajapinnalla voi asettaa ogg äänen voimakkuuden(volume)
2.2.5	äänirajapinnalla voi asettaa ogg äänen toistokorkeuden(pitch)
2.4	äänirajapinta voi ladata WAV ja OGG tiedoston pakatusta hakemistosta
2.5	äänirajapinta voi ladata WAV ja OGG tiedoston hakemistosta
3	äänirajapinta toimii Windows 7, Vista ja XP alustoilla
3.1	äänirajapinta voi toistaa WAV tiedostoja
3.1.1	äänirajapinta toistaa wav tiedoston
3.1.2	äänirajapinnalla voi pysäyttää wav tiedoston toiston
3.1.3	äänirajapinnalla voi tauottaa wav tiedoston toiston
3.1.4	äänirajapinnalla voi asettaa wav äänen voimakkuuden(volume)
3.1.5	äänirajapinnalla voi asettaa wav äänen toistokorkeuden(pitch)
3.2	äänirajapinta voi toistaa OGG tiedostoja
3.2.1	äänirajapinta toistaa ogg tiedoston
3.2.2	äänirajapinnalla voi pysäyttää ogg tiedoston toiston
3.2.3	äänirajapinnalla voi tauottaa ogg tiedoston toiston
3.2.4	äänirajapinnalla voi asettaa ogg äänen voimakkuuden(volume)
3.2.5	äänirajapinnalla voi asettaa ogg äänen toistokorkeuden(pitch)
3.4	äänirajapinta voi ladata WAV ja OGG tiedoston pakatusta hakemistosta
3.5	äänirajapinta voi ladata WAV ja OGG tiedoston hakemistosta

```

# Makefile for AUDIO library
# define a list of pkg-config packages we want to use
pkg_packages := gtk+-2.0
PKG_CFLAGS := $(shell pkg-config --cflags $(pkg_packages))
ADD_CFLAGS := -Wall -c -DRELEASE -DNDEBUG -fvisibility-inlines-hidden
CPP := g++

#check target info
OUTPUT_FILE := libaudio.a
OUTLIB = ""
ifeq ($(SBOX_UNAME_MACHINE),i486)
    OUTLIB := ../../../../lib/maemo/x86/$(OUTPUT_FILE)
else
    ifeq ($(SBOX_UNAME_MACHINE),arm)
        OUTLIB := ../../../../lib/maemo/armel/$(OUTPUT_FILE)
    endif
endif
endif

# combine the flags
CFLAGS := $(PKG_CFLAGS) $(ADD_CFLAGS) $(CFLAGS)
DIR := ../../source
INCLUDES := -I../../include -I../../HAL/include -I../../render/include -I../../external/slmath/include
-I../../external -I../../core/include -I../../external/libogg-1.2.0/include
-I../../external/libvorbis-1.3.1/include
OBJ := Audiostream.o\
      Wavstream.o\
      Oggstream.o\
      SoundManager.o

.PHONY: all
all: $(OUTLIB)

$(OUTLIB): $(OBJ:.o=)
    $(RM) $(OUTLIB)
    ar rcs $(OUTLIB) $(OBJ)

$:
    $(CPP) $(CFLAGS) $(INCLUDES) $(DIR)/$@.cpp -o $@.o

.PHONY: clean
clean:
    $(RM) $(OBJ)
    $(RM) $(OUTLIB)

```

```

# Makefile for audioTest
pkg_packages := gtk+-2.0
PKG_CFLAGS := $(shell pkg-config --cflags $(pkg_packages))
PKG_LDFLAGS := $(shell pkg-config --libs $(pkg_packages))
ADD_CFLAGS := -Wall -c -DRELEASE -DNDEBUG
KAJAK_LIBS := -lframework -lrender -lcore -lHAL -lslmath -lX11 -lode -lGLES_CM -laudio
-lopenal -lalut -lvorbisenc -lvorbisfile -lvorbis -logg
CPP := g++
LD := g++

KAJAK_LIBS := -L../../../../lib/maemo/armel/ $(KAJAK_LIBS)

# combine the flags
CFLAGS := $(PKG_CFLAGS) $(ADD_CFLAGS) $(CFLAGS)
LDFLAGS := $(PKG_LDFLAGS) $(LDFLAGS)
PROGRAM := Test
DIR := ../source
INCLUDES := -I../../../../HAL/include -I../../../../external -I../../../../render/include
-I../../../../external/slmath/include -I../../../../external/ode-0.11/include -I../../../../core/include
-I../../../../framework/include -I../../../../audio/include -I./
-I../../../../external/libogg-1.2.0/include -I../../../../external/libvorbis-1.3.1/include
OBJ := MyApp.o

.PHONY: all
all: $(PROGRAM)

$(PROGRAM): $(OBJ:.o=)
    $(LD) $(PKG_LDFLAGS) $(OBJ) $(KAJAK_LIBS) -o $(PROGRAM)

%:
    $(CPP) $(CFLAGS) $(INCLUDES) $(DIR)/%.cpp -o %.o

.PHONY: clean
clean:
    $(RM) $(OBJ)
    $(RM) $(PROGRAM)

```