FAKULTÄT FÜR !NFORMATIK

MeXX
A Virtual World for Exploring Musical Concepts

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Computergraphik/Digitale Bildverarbeitung

eingereicht von

Michaela Fruhwirth, BSC
Matrikelnummer 0208553

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung:
Betreuer: ao.Univ.Prof. Dr. Andreas Rauber

Wien, 22.März 2010

Michaela Fruhwirth | Prof. Dr. Andreas Rauber

Erklärung zur Verfassung der Arbeit

„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit einschließlich Tabellen, Karten und Abbildungen, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe."

Michaela Fruhwirth
Rosensteingasse 7-9/2/14
1160 Wien

Danksagung

An erster Stelle möchte ich mich recht herzlich bei ao.Univ.Prof. Dr. Andreas Rauber für seine Betreuung bedanken und dafür daß er mir ermöglicht hat diese Diplomarbeit umzusetzen. Ein weiterer Dank geht auch an Rudolf Mayer für die Bereitstellung der Datenbank und für hilfreiche Tips bei der Implementierung. Ein großes Dankeschön gilt auch meiner Familie, die immer hinter mir gestanden ist und mich zu jeder Zeit unterstützt hat. Im speziellen möchte ich mich auch bei meinem Vater bedanken, der mir jederzeit mit Rat und Tat zur Seite gestanden ist.

Zusammenfassung

Musik an sich und ihre Vielfalt an Eigenschaften inspiriert Wissenschaftler neuartige Software Systeme zu entwickeln welche Musik in unterschiedlichster Art und Weise zugänglich machen. Die hier vorgestellte Diplomarbeit konzentriert sich auf moderne Musik Systeme welche auf eine traditionelle, hierarchische Anordnung von Musik verzichten um diese zu durchforschen und zu erleben. 2- und 3 dimensionale Visualisierungssysteme aus den Bereichen Music Information Retrieval, Virtuelle Umgebungen und Musik-basierten Spielen, wurden dazu im theoretischen Teil analysiert und evaluiert.

Der praktische Teil präsentiert die Entwicklung einer virtuellen Musikwelt welche eine sowohl audielle als auch visuelle Präsentation von Musikstücken in einem virtuellen Umfeld – der 3D Welt – ermöglicht. Diese 3D Welt besitzt die Möglichkeiten, Musik in einer virtuellen Umgebung zu konsumieren als auch als Spiel das Verständnis des Benutzers für musikalische Konzepte zu trainieren.

Musikstücken lassen sich grundsätzlich immer unterschiedlichste Eigenschaften zuordnen. Die Grundlage der 3D Welt bilden vier spezielle Eigenschaften; das *Genre*, die *Stimmung*, das *Tempo* und das *Erscheinungsjahr*. Diese sind mit den Musikstücken in einer Datenbank gespeichert welche noch zusätzliche Informationen über Künstler und Alben enthält. Die Musikstücke werden visualisiert indem sie als Objekte in der 3D Welt dargestellt werden. Diese wird gebildet indem zwei Eigenschaften der Musikstücke den beiden Basisachsen der 3D Welt zugeordnet werden. Die Musikstücke sind demzufolge in der 3D Welt durch diese beiden Eigenschaften eindeutig positioniert. Optional lässt sich eine dritte Eigenschaft für die farbliche Darstellung der Objekte verwenden. Durch Navigation besteht die Möglichkeit die Welt zu durchwandern und die Musikstücke anzuhören. Ein 3D Sound System gibt die Musikstücke als räumlichen Sound wieder mit der Möglichkeit zur Überblendung mehrerer Musikstücke, Richtung und Lautstärke sind dabei von der Position des Betrachters abhängig.

Abhängig von der Fülle an Informationen in der Musikdatenbank lassen sich unterschiedlichste Erkenntnisse in unterhaltsamer Weise gewinnen. Dies kann beispielsweise die Entwicklung eines Genres über die Jahre oder auch die Verteilung von Musikstücken eines einzigen Künstlers in der 3D Welt sein. Die Spieloption erfordert, eine bestimmte Anzahl an Musikstücken korrekt in der 3D Welt einzuordnen. Da die Welt basierend auf den Eigenschaften der Musikstücke aufgebaut ist, wird der Spieler angeregt, diese ausschließlich durch anhören zu erkennen. Das Spiel wurde mit dem Ziel entwickelt das Gespür des Benutzers für Musik zu trainieren. Das System hat den Namen MeXX und ist als Single-User Java Applikation konzipiert. Die eigentliche 3D Welt wurde mit Hilfe der Java3D API realisiert. Der 3D Sound wurde durch die *PointSound* Implementierung der Java3D API und dem JOAL-Mixer als geeignetem Audio Mixer bewerkstelligt.

Abstract

Music itself and its richness on features inspire scientists to develop novel software systems that deal with music in a wide variety. This master thesis focuses on modern music visualization systems which go beyond traditional directory structures to organize and represent music. They are actively being implemented by researches to provide users with a novel interface in music browsing and experiencing in a 2- or 3-dimensional fashion. Music systems out of the areas Music Information Retrieval, Virtual Environments and Music-Based Games have been analyzed and evaluated in the theoretical part.

The practical part of this thesis presents a virtual music world which offers both an aural and a visual presentation of music pieces inside a virtual environment – the 3D world. It has been realized as a Java implementation with the options to consume music in a virtual environment and to train a user's sense of music in an entertaining way by a game.

Generally a large variety of features can be attached at all times to music pieces. The foundation for the 3D world is formed by four particular features; the *genre*, the *mood*, the *tempo* and the *year* of publication. A database stores those features including the music pieces as well as other information about artists and album. A selective access to the database is enabled by a filter.

The music pieces are visualized as they are displayed as objects in the 3D world. This world is formed by two selectable features of the music pieces being mapped to the two axes of the 3D world spanning the ground. As a result the music pieces are unambiguously positioned in the 3D world by these two features. Optionally a third feature can be used for color encoding of the objects. The third spatial dimension of the 3D world is used only for display purposes. Integrated navigation enables to visit the 3D world and to look at and listen to the music pieces. The implementation of a 3D sound system enables the playback of the music pieces as spatial sound including cross-fading of several music pieces, direction and volume are thereby depending on the visitor's position.

Depending on the information a music repository contains, a variety of perceptions of and insights into music may be entertainingly obtained. This can be for instance the development of genres over the years as well as monitoring the distribution of several artworks of a single artist inside the 3D world. The game option presents a subset of music pieces that need to be assigned at correct positions within the 3D world. Since the features of those music pieces need to be identified by listening to conclude their position, the game challenges the appreciation for music again in an entertaining way.

The system is given the name MeXX and is a single-user Java application. The 3D world itself and its contents have been developed using the Java3D API. The 3D sounds are established through Java3D`s *PointSound* implementation in combination with the JOALMixer as an adequate audio device.

Content

List of Tables

List of Figures

1 Introduction

Music is art. The composition of musical works, the selection and arrangement of musical instruments and the final performance of a musical work are all kind of arts in producing music. They have a high degree of freedom and the results depend on individual tastes and interpretations.

Music is ubiquitous. Music is being targeted consumed by human beings on their personal audio playback devices, on radio stations, video clips, on concerts, bars and clubs or in music stores. Music is being heard as background sounds in shops, restaurants, or sport events. Furthermore, music is used in the entertainment domain and occurs in movies, commercials and shows to force emotions and impressions.

Music is inspiration

Nowadays we are not limited by physical and analog restrictions to music performance. The digital era provides us with new possibilities in exploring and studying music. Music itself and its richness on features inspire modern scientists to develop novel software systems that deal with music in a wide variety. This ranges from digital sound analysis and exploration to artificial understanding of music by computers to music visualization systems.

The virtual dimension has captured us already and that fact in turn opens up a variety of possibilities in designing software systems to produce 3-dimensional data representations. Application areas are scientific data visualizations, virtual workspaces and universities, virtual online communities and museums as well as 3-dimensional representation systems for music.

This master thesis focuses on modern music visualization systems which go beyond traditional directory structures to organize and represent music. Such music systems are actively being implemented by researches to provide users with a novel interface in music browsing and experiencing in a 2- or 3-dimensional fashion.

The aim of this master thesis is the design of a 3D music world for music experience that provides spatial sound capabilities and meets two purposes: a customizable visualization layout to present a music repository and a game environment setup to interactively deal with music. The system is called MeXX and for the implementation Java3D was chosen since it provides functionalities to create a 3D world with spatial sound sources. The data base for MeXX is a music repository that contains information about music pieces, albums and artists. Music pieces are thereby the central element and described by features. The features *genre*, *mood*, *tempo* and *year* were chosen as candidates to define the position of a music piece within the 3D world. Two out of these features will serve as an arrangement base for the music pieces in the 3D world. Therefore these two features are mapped on two dimensions of the 3D world. These dimensions define directions in space and in MeXX they follow the local x-, and z-axis which span the ground of the 3D world. After the world is built, navigation through it to explore and examine the music repository is possible. Every music piece in the 3D world plays it's corresponding audio file as spatial sound, established by Java3D`s *PointSound* implementation. Depending on the information a music repository contains, a variety of perceptions and insights into music may be entertainingly obtained. This can be for instance the development of genres over the years as well as monitoring the distribution of several artworks of a single artist inside the 3D world.

MeXX may also be initialized as a 3D game environment with an official goal to collect scores and an unofficial one to train the user's sense and appreciation for music. Therefore the user has the option to select two features to build up the 3D world, which implies that the user will be challenged on those two features. The user is presented a number of songs that have to be positioned correctly within the 3D world. Thereby it is necessary to identify the features of those songs to conclude the correct position, whereas the only thing the user can do with the songs is to listen to them. To estimate the correct position the user can navigate through the 3D world to aurally check the locations of the remaining music pieces which reside at their appropriate positions. At the end of the game the correct positions and the obtained score are presented. Utilizing MeXX as game forces the user to attentively listen to the songs to identify their features and thus a non-obvious learning process is activated as a natural benefit.

The master thesis is structured as follows. Chapter 2 provides an overview about the related work. This includes the research areas Music Information Retrieval, Virtual Environments in general, Virtual Music Environments and Music-Based Games. In Chapter 3 the concept of MeXX is presented in detail as well as the implementation. In Chapter 4 MeXX is presented in all its facets and features and screenshots of the final application as well as application scenarios are presented. Chapter 5 finally draws some conclusions and discusses future work. The websites mentioned in this document were last accessed in March 2010.

2 Related Work

In this chapter related works in the domains of Music Information Retrieval (MIR), virtual environments and music-based games are presented. Starting with an introduction, Chapter 2.2 follows with an overview about music information and MIR techniques. Furthermore visualization techniques in MIR systems are presented. Chapter 2.3 concentrates on virtual reality and its application areas according to 3D virtual environments. In Chapter 2.4 music-based games are discussed and example games are presented. Finally Chapter 2.5 presents a selection on existing 3D music environment systems and music installations in the real world which all focus on an audible and novel perception of music.

2.1 Introduction

Music in general, Music Information Retrieval (MIR), virtual 3D environments, music-based games and 3D music environments served as an inspiration source for this master thesis. MIR has a broad range of research tasks - audio analysis, audio segmentation, feature extraction from audio signals, automatic and a novel organization of music libraries and automatic classification of music pieces into categories are just a few examples. The focus within this thesis lies on the examination of different visualization layouts that MIR systems provide for browsing and representing music libraries.

Virtual environments are introduced in several application areas ranging from online communities to virtual museums and from cultural heritage to virtual workspaces and scientific information spaces. All aim for a 3D representation of particular information a user can browse through space and perceive it more intensively compared to a 2D representation.

Music-based games are a specific style of games designed around an auditory experience. The player is forced to deal with music in an intensive way – this ranges from rhythm, voice and beat reproduction to physical performances to the creation of own sounds and melodies within music-based games. Like common games, also music-based games have a goal for motivation – the better the player reproduces the given music the more score is obtained and the game level increases.

Music environments (either implemented as 3D or hybrid environments or installed in reality) are being developed to provide users with novel perception and interaction possibilities on music. 3D music environments are for instance concert simulations, systems for collaborative music production, environments that visualize music libraries or 3D music stores and dance clubs.

The next chapter is dealing with MIR and illustrates several fundamentals of music. Music representation forms are depicted and visualization capabilities in MIR systems are presented.

2.2 Music Information Retrieval

Music Information Retrieval (MIR) is an interdisciplinary and young research area which began to grow drastically in the late 1990s. It is devoted to the development of technologies and tools for an efficient access to music libraries. Every research domain within MIR has its own set of goals, techniques, research questions and evaluation paradigms; audio engineers work on signal processing, musicologists are dealing with symbolic representation issues, computer scientists work on pattern matching techniques and librarians' work is related to bibliographic description concerns. But also information science, music theory, law and business are interested in MIR tools for investigating music (Downie, 2003).

MIR techniques are used to go beyond traditional directory structures of music libraries and support non-specific searches of music. Users may want to find particular music for a specific situation or mood or want to reorganize their music collection in terms of "happy music", "romantic music", "party music", "fast music" or other desired aspects, which is difficult to establish in hierarchical structures. Furthermore if the music collection grows, unknown or forgotten music may increase in hierarchical music structures.

Lee et al. (2004) present an empirical evaluation of music information needs, uses and seeking behaviors of human beings. The aim of this study was to provide an empirical foundation for the development of MIR and Music Digital Library (MDL) systems and prevent false assumptions in designing specific technologies. 427 people from the University of Illinois at Urbana-Champaign (UIUC) were interviewed. Table-1 depicts the top three answers of this study.

Reasons for music search	Music information requirements
Users seek music to listen for entertainment (94.5%)	Title of a musical work (90%)
Users seek music information to build or assist music collections (89.1%)	Lyrics of a song (81.0%)
Users seek music information for verifying or identifying works, artists, and lyrics (73.9%)	Artist information (74.6%)

Table-1: Reasons for music searching and music information needs (Lee et al., 2004)

Orio (2006) classifies possible users of MIR systems into three groups according to their knowledge about music theory and practice in

- Casual users
- Professional users and
- Music theorists, musicologists and musicians

Casual users intend to listen to music for enjoyment or for collecting. They have no theoretical background of music and are not able to describe their musical information requirements in terms of musical dimensions – the *query by example* approach is appropriate for this users.

Professional users have usually a good knowledge about the music domain and its language. They are able to describe in detail and in a more abstract level their requirements. Due to their professional activities they probably need access to music collections including radio and television broadcasts to search for particular soundtracks, commercials, news stories, documentaries, or for the organization of live performances and concerts.

Music theorists, *musicologists* and *musicians* are interested in analyzing or studying a musical document rather than listening to it. They are familiar with the music language in all of its dimensions including the terminology to describe music and metadata associated with a musical work.

2.2.1 Fundamentals of Music Information

The culture of music ranges from the common practice area (the periods of baroque, classical music and romance, 1600-1900, which is from Bach to Brahms) to modern rock and pop music to improvised jazz, electronic art music and further to Asian music and performances of Indian ragas. The perception, appreciation and experience of music exist in the mind of the perceiver and depend on the mood, situation and circumstances. Music can be an object of study (through music scores or sheets or through live recordings), slightly perceived as background music (during housework, sports, studying, working or waiting) or is experienced through a continuation of familiar traditions like the singing of hymns, camp songs and nursery songs or as a means of religious expression (Downie, 2003).

Music Information

Music is the art of organizing interplay of sound events, in a way to produce a melodious arrangement. Downie (2003) proposes to conceive music by seven facets. These facets are fundamental and relevant content descriptors of music and listed below.

- **Pitch Facet –** is the fundamental frequency of a sound – i.e. the number of oscillations per second. It is said to range from low to high. For example the standard pitch or concert A has a frequency of 440 Hz and is used for the adjustment of instruments. A pitch can be graphically represented by either the vertical position of a note on the staff, by its name (e.g. A, D#, Eb), by its scale degrees (e.g. I, II, III, …), by the pitch class numbers (e.g. 0, 1, 2, …) or by solfége (e.g. do, ré, mi, …).

- **Temporal Facet –** is the information about the duration of a musical event. This includes tempo indicators, meter, pitch duration, harmonic duration and accents. Together these five elements constitute the rhythm. The temporal information can be absolute (e.g. a tempo indication of bpm[1] = 80), general (e.g. *adagio*, *presto*) or relative (e.g. *più lento*, *accelerando*).

- **Harmonic Facet –** harmony occurs when two or more pitches sound at the same time – this is also called polyphony. When only one pitch sounds at a time it is called monophony. Harmonic events are represented for example by chords – on the staff indicated by notes on top of each other - or by names like C, Am or through the Roman numeral notation like I-IV-V-I.

- **Timbral Facet –** is the sound characteristic of a specific instrument – the sound of a flute has a different timbre than the sound of a piano. This aural phenomenon is also said to be the color of a tone.

- **Editorial Facet –** are performance instructions like fingering, ornamentation, dynamic instructions (e.g. *pianissimo*, *piano*, *forte*, *crescendo*), articulation (e.g. *legato*, *staccato*) or slur.

- **Textual Facet –** are for example the lyrics of a song, chorals, arias, hymns or symphonies.

- **Bibliographic Facet –** includes additional descriptive information about a musical work. This includes a work's title, composer, arranger, editor, lyric author, publisher, edition, publication date, discography and performer.

Furthermore the *orchestration* (selection of instruments employed within a musical work), the *acoustics* (dealing with perception of sound and its quality, including room acoustics, background noise, post-processing and filtering) and the *melody* (a recognizable sequence of tones performed with a similar timbre – for example the singing voice) are relevant facets of music (Orio, 2006).

Figure-1: Music score excerpt of Mozart's opera *Cosi Fan Tutte*

Figure-1 shows an excerpt of the musical score (also referred to as sheet music) of Mozart's opera *Cosi Fan Tutte*, debut-performed in 1790 in Vienna. The orchestration consists of flutes, oboes, clari-

[1] bpm – beats per minute

nets, horns, bassoons and the soprano voice of Anna Despina singing the melody. The excerpt depicts the duration and pauses of the pitches (i.e. the musical notes) for each instrument and starts with a medium tempo of *andante* having a meter of 6/8. Loudness indicators ranging from loud (*f*) to low (*p*) are used as well as slur which indicate a smooth transition between the included notes.

Music Representations

Music can be represented and described in different ways whereas the choice of representation depends on the system and the desired information (Downie, 2003):

- Audio representation
- Symbolic representation

Audio representations include analog and digital live performances and recordings like LP's, mp3 files, CD's and tapes. Nowadays modern formats contain digital coded waves of audio signals, where compressed and uncompressed formats are distinguished. Uncompressed formats are based on the pulse code modulation (PCM) representation of an audio signal; examples are the `.aiff` and `.wav` formats. Since uncompressed formats require great storage capacity they are often being compressed; examples are the `.mp3`, `.wma` and `.ogg` formats.

Symbolic representations include printed notes, scores, text and specific computer formats like musical instrument digital interface (MIDI), GUIDO music notation format, kern and notation interchange file format (NIFF).

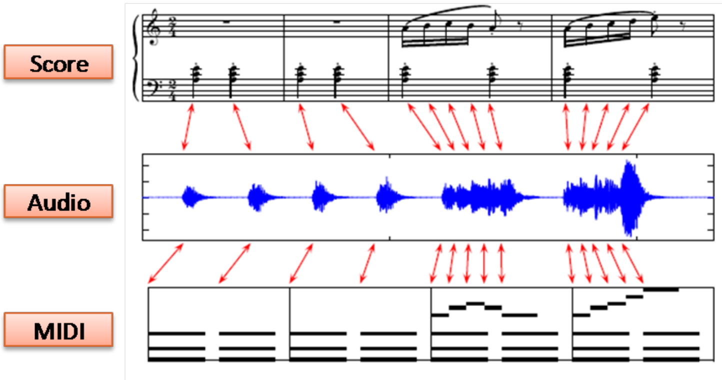


Figure-2: Relation between score, audio and MIDI information

Figure-2 is adopted from Müller (2007) and sketches the relationship between the symbolic score representation (top), the audio signal (mid) and the MIDI format (bottom) of a music piece. The arrows indicate the corresponding music events in the different representations.

Working with Music Representations

Depending on the subject of interest and the underlying music format there are several approaches to music processing whereas the main part of feature research and extraction is done with the audio form of music (Orio, 2006).

Bainbridge et al. (1999) introduced prototypes to collect, access and present digital music libraries. The prototypes provide the user with different querying functionalities to receive a particular musical work. The underlying music database consists of recorded music, MIDI files, printed sheet music and textual information like biographies. To automatically convert the printed sheet music into a digital computer-readable form the *optical music recognition* (OMR) technique is used. The principle of this technique is shown in Figure-3, the scanned sheet music is converted into a digital symbolic format by use of image processing techniques.

Figure-3: The principle of Optical Music Recognition (OMR)

For searching and browsing the music library, text-based querying is supported where the user simply searches for a specific musical work by use of keywords. In addition melody-based querying is implemented where the user can sing, hum or whistle a remembered part of a song into a microphone. The recorded audio data are afterwards converted into a symbolic representation and the system searches for a similar melody in the music database; the matching process is based on musical notes. Instead of singing, the user can also enter a remembered part of a song using a computer keyboard.

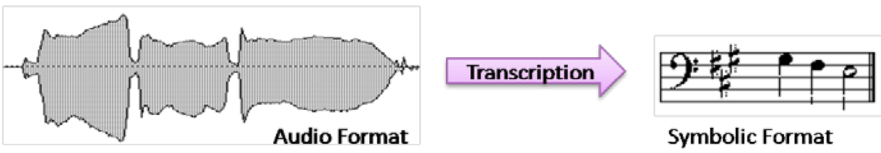


Figure-4: Transcription from audio to symbolic format

Figure-4 displays the principle of transcribing a musical sequence from its audio format into its symbolic format. Furthermore the system supports a combination of text-based and melody-based search techniques. After searching the user is provided by the system with music data including text (title, artist, or composer), the MIDI file and the symbolic music notes received directly from the MIDI file.

Today the *International Society for Music Information Retrieval* (ISMIR)[2] and its conferences have become an informative exchange platform of related research done on MIR. Content-based MIR techniques work with an audio representation of music pieces. The idea behind this approach is that a music piece can be described by a set of features which are directly computed from its content (Orio, 2006). Once adequate features are determined they can be used to automatically classify music pieces into defined categories like artist, genre, rhythm or mood making use of modern machine learning techniques. The advantage in using content-based approaches for music classification is that they do not depend on manual annotations from human beings and are therefore cheaper since the computation is done automatically (Pampalk et al., 2006). Through content-based MIR, it is possible to automatically cluster and organize music libraries regarding to specific desires and visualization layouts; Chapter 2.2.2 concentrates on novel user interfaces obtained by MIR. Furthermore automatic music recommendation systems and automatic playlist generation techniques make use of MIR. The automatic recognition of the orchestration or the rhythm of a song, the extraction of the main melody or an automatic recognition of chord sequences within a music piece are further research topics within MIR (Orio, 2006).

Metadata of Music

In addition to symbolic and audio representation, music can be described by metadata. Metadata can be factual information like artist, album, year or publication as well as track title and duration, but can be also subjective or cultural determined information like mood, emotion, genre and style; usually being annotated by human beings.

Metadata of audio files are useful for searching and indexing music and sound databases. They can be used for system evaluations or to generate training data sets for machine learning. A popular project aimed at music annotation is *Pandora's Music Genome Project* [3]; in the year 2000 a group of approximately 50 expert music reviewers got together to create a comprehensive analysis of music. A song was examined by all experts and annotated using structured vocabularies of between 150-500 tags, describing how the song sounds like – i.e. its melody, harmony, rhythm, instrumentation, orchestration, arrangement, lyrics, vocal and singing harmony.

All Music Guide (AMG)[4] provides as well a comprehensive music reference source. The project was founded in the year 1990 with the aim to help consumers to navigate within the increasingly complex world of music. The website provides information about music of four categories: metadata (includes

[2] Homepage of ISMIR: www.ismir.net/
[3] Homepage of Pandora's Music Genome Project: www.pandora.com/mgp.shtml
[4] Homepage of All Music Guide: www.allmusic.com

facts about an artist or album like titles, genre and release date), descriptive content (includes deeper details like moods, instrumentation), relational content (like similar artists and major influencers) and editorial content (including biographies and reviews).

Last.fm[5] is a popular internet radio service which besides listening to music provides information about artists, albums, tracks, similar artists, charts and statistics about music. Visitors are automatically provided with similar music of music they are listening to and in addition they can actively contribute in changing music's metadata by rating, describing or recommending a song. In addition *Last.fm* allows anyone to use its music and metadata.

But since the cost of obtaining a comprehensive set of manual annotations is high, further possibilities developed to manually annotate music pieces are annotation games. The aim is that people will voluntarily play them and produce useful metadata as a by-product. Mandel et al. (2007) present the web-based game *Major Miner*[6]. Participants are given one randomly selected clip and they have to describe it with a word or phrase like pop, piano, sad. Depending on how much other users have used this tag for the song before, the player scores points and the system stores the retrieved metadata. Law et al. (2007) present *Tag a Tune*, a web-based game where randomly paired players are given several sounds (which can be music clips, rhythms, effects, ambience noise or speech). Then they are challenged to guess what their partner is thinking about the sound. A description becomes an official tag when it is agreed upon a specific number of people. In the *Listen Game* (Turnbull et al., 2007), a group of users is presented with a song and a list of structured tags. The players have to choose the best and worst tags for describing the song. The larger the number of people which agree on a tag, the stronger is the songs (positive or negative) association to the tag.

2.2.2 Visualizations in Music Information Retrieval Systems

In this chapter a selection of MIR systems is presented which provide innovative user interfaces to explore music repositories. Their interfaces differ from common music libraries like the *Windows Media Player* or *Winamp* where music is organized in predefined hierarchical structures. The below mentioned systems use either content-based audio analysis or metadata for visualization tasks.

The Islands of Music

Islands of Music (Pampalk, 2003) is a visualization tool to browse music pieces in terms of their similarity. Rhythm patterns are used to define similarities between songs and Self Organizing Maps (SOM) are trained to visualize these similarities on a 2D map. For an intuitive visualization of the

[5] Homepage of Last.fm: www.last.fm
[6] Homepage of Major Miner: www.majorminer.org

resulting genres, the island metaphor is used; similar pieces of music are positioned close to each other and thus form the music islands. Figure-5 (left) shows a screenshot of the system: the user can browse a music collection by explore one island after another and listen to songs by clicking on its representation. The upper left island represents classical music unlike the smaller island near the lower left corner which includes more aggressive music like metal and rock.

The Map of Mozart

The *Map of Mozart* (Mayer et al., 2006) visualizes the complete music works of Wolfgang Amadeus Mozart. The music collection consists of 2.442 musical documents which are manually classified into 17 different categories like operas, concertos, symphonies, canons or violin sonatas. Additionally the music pieces are clustered in terms of their rhythmic similarities and visualized using a mnemonic SOM. With this technique it is possible to project the information on the silhouette of Mozart's head or other non-rectangular maps like countries or geometrical figures.

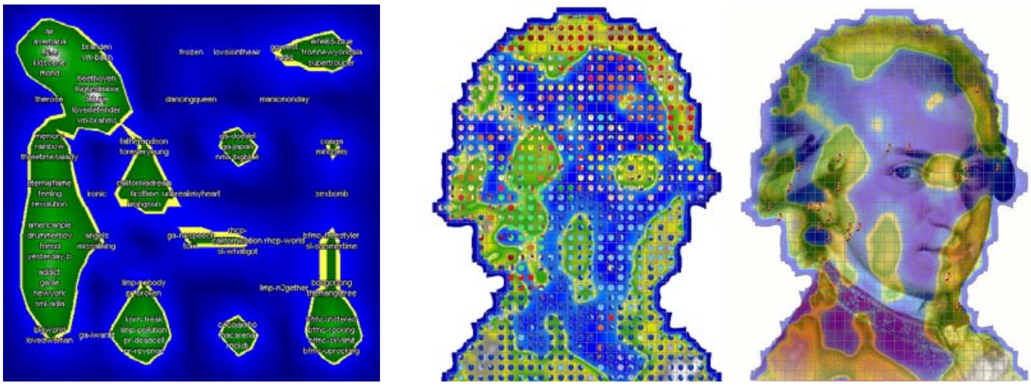


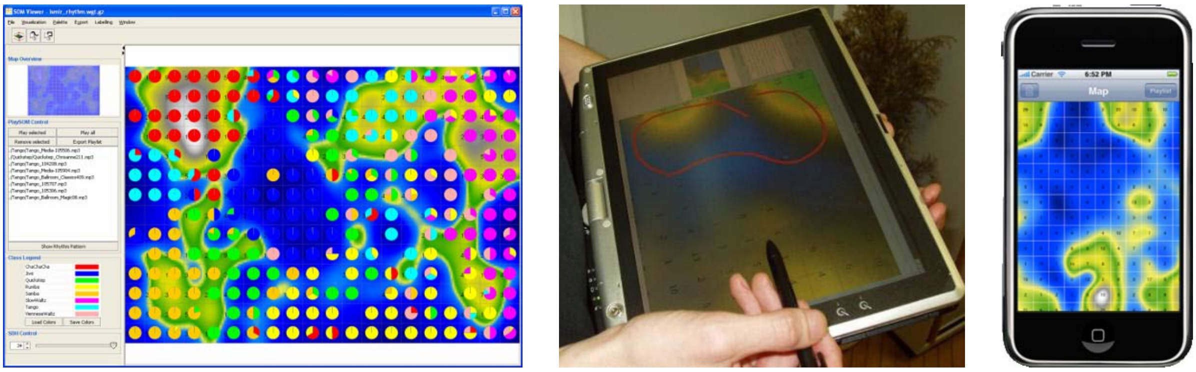
Figure-5: *Islands of Music* (left), the *Map of Mozart* (mid & right)

The user can switch between several visualization layouts combinable with the categorical information when exploring the musical works. Figure-5 (mid) shows one possible visualization layout of the system; the similarities are represented through the *Islands of Music* approach, and each song is additionally color encoded according to its category. For example concertos are colored through red, songs are colored dark green, and operas are blue. Figure-5 (right) shows a semitransparent view of Mozart's head combined with the *Islands of Music*. The user is also provided with additional information about songs including the title and the rhythm patterns by clicking on them.

The PlaySOM and PocketSOM Player

The *PlaySOM* and *PocketSOM* player (Neumayer et al., 2005) are another two visualization interfaces to browse music collections based on the *Islands of Music* approach. The user can additionally generate customized playlists by drawing trajectories on the music map; the songs located at this path thus form the playlist. A zooming mechanism provides the user with more detailed information about the tracks and a rectangle selection can be performed to only play tracks of a specific area. Also different visualization layouts can be combined: Figure-6 (left) shows the genre distribution of the music pieces besides the music islands.

The *PlaySOM* application was designed for desktop computers, laptops or tablet PC's. In Figure-6 (mid) the user creates a playlist in evidence through the red path on the tablet PC. The *PocketSOM* player is a lightweight version of the *PocketSOM* application and can be used for mobile devices as shown in Figure-6 (right).



<u>Figure-6:</u> *PlaySOM* interface (left & mid), *PocketSOM* Player (right)

The SoniXplorer

Lübbers (2005) proposes two user interfaces to explore music collections: The *Sonic Radar* and the *Sonic SOM*, both using spectral characteristics for similarity computation of the music pieces. Within the *Sonic Radar* interface the music collection is clustered using the k-means clustering approach. For each cluster a prototype song which has the closest distance to the center of its cluster is selected. The prototypes are placed on a circle and the user is represented in the center of this circle. By changing its focus of perception, i.e. rotating the circle, the user can control the playback of the prototype songs. Focusing on only one song or integrating surrounding songs by changing the angle is therefore possible. Considering stereo speakers or headphones the implementation either pans the audio signal on the left, the right or both channels (i.e. direct focus on one song).

The *Sonic SOM* implementation uses a SOM for mapping the songs on a 2D map. The environment around the user's standpoint is partitioned into k direction classes and the closest songs of each class are considered to be played back. Like the *Sonic Radar* the user can change the focus by rotating the SOM map to change the loudness and inclusion of neighboring songs.

The Music Rainbow

The *Music Rainbow* (Pampalk et al., 2006) is a visualization tool to discover artists by their similarity. The system is shown in Figure-7 (left), the rainbow consists of eight colored concentric circles, each representing a different music style. Inside the circle high-level genre terms like rag, jazz or soul are depicted which describe the rough structure of the rainbow; each circle corresponds to one level. More specific information about the rainbow circles is located outside them and is descriptive text like female, piano, orchestra or club.

Each rainbow is segmented into equally spaced arcs and contains associated artists labeled with keywords to describe them. Content based analysis of the audio signals is performed as well as information about artists is extracted from the web to compute similarities between artists. A *Griffin PowerMate Knob* is utilized as input device to rotate the rainbow and to explore the artists.

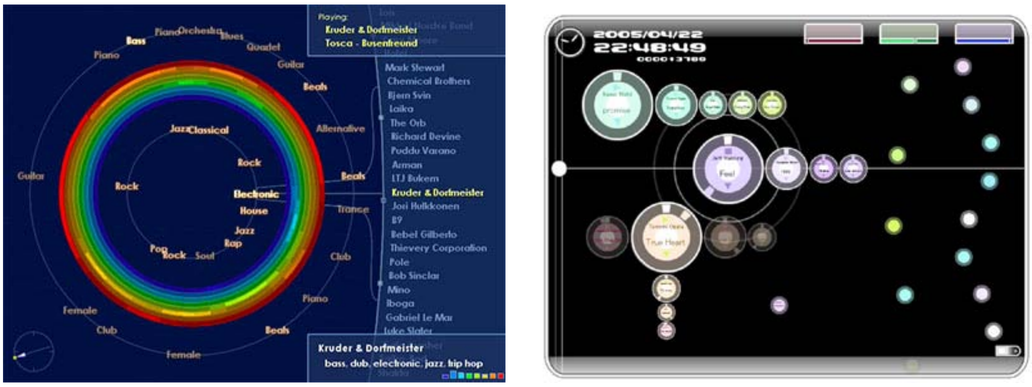


Figure-7: The Music Rainbow (left), the Musicream (right)

The Musicream

The *Musicream* introduced by (Goto et al., 2005) is a dynamic music playback interface to retrieve similar music pieces to those preferred by the user. The interface, depicted in Figure-7 (right), consists of three boxes at the top, the so called *music-supply tabs*, that hold a specific color to reflect the mood or feeling of a music piece. Within the *Musicream* four different functions are implemented: the *music-disc streaming* function streams down the music discs one by one from the *music-supply tabs* and the user can select and peruse them by simply clicking on them for listening. The *similarity-based sticking* function sticks together similar music pieces in terms of their mood by simply dragging one music piece over the others to (magnetically) catch similar ones. Through the *meta-playlist* function the user can arrange groups of music pieces and try out different playback orders. The *time-machine* function stores all user operations and screen changes to let the user turn back to a previous playlist or listening state.

Artist Map

van Gulik et al. (2004) present the *Artist Map*, a user interface especially meant for small devices like mobile phones. The user can customize the view of an artist map by mapping the attributes mood, genre, year and tempo either to the vertical or horizontal axes and makes use of color coding. Figure-8 (left) shows one possible layout with the artists represented by the circles. Five different tempo ranges and six year ranges were chosen as vertical respectively horizontal attributes to position the artists on the map. Furthermore the tempo is color encoded from blue to red (very slow to very fast). In addition to a customized view, the user can search for artists by the included hierarchical metadata.

Visual Collaging

Bainbridge et al. (2004) present a music browsing system based on visual collaging. The visualization is based on the *laid back* approach where little or even no control of the user is needed. Besides simple browsing by an artist's or song's name, the user can browse a library by observing a set of faded images. Two images like an album cover, publicity shots of an artist or photographs of vinyl discs represent a particular song.

Figure-8: The Artist-Map (left), Visual Collaging (right)

The images consecutively fade from the front to the back and the user perceives the process of music browsing as visual impressions. Clicking on an image pops up a new window that shows the related images with further information about the song and corresponding artist and provides a media player to listen to the music.

A Disc-, Rectangle- and Tree-Map Visualization

Torrens et al. (2004) introduce metadata-based disc-, rectangle- and tree-map visualizations that are aimed to provide an overview about the content of a music library. Music tracks have the attributes genre, artist and year. Within the disc-visualization the whole library is represented on a disc chart. Figure-9 (left) shows a screenshot, the disc is divided into different genre sections and again split up in sub-sectors representing the associated artist as points. The ratio of the disc depicts the year whereas at the center older songs are located ranging to more recent tracks at the outside of the disc.

Furthermore color supported interaction scenarios like navigation, zooming, standard searching and playlist management are implemented; for example the most and least played songs can be color highlighted. The rectangle visualization is similar to the disc one but the year is represented by the vertical axes and the genre by the horizontal axes. The tree-map visualization as shown in Figure-9 (right) splits up the whole library into genres, subgenres and artists whereas tracks are not directly visible.

This visualization layout is thus more suitable to give an overview about the number of tracks assigned to an attribute compared to those highlighted by color encoding.

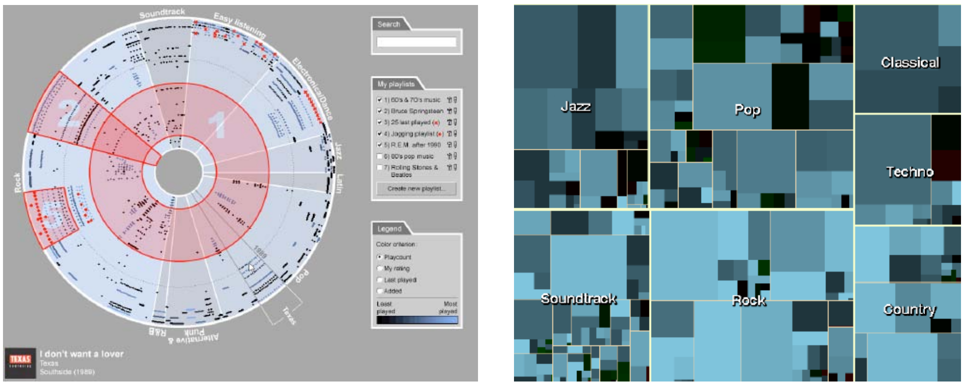


Figure-9: Disc visualization (left), tree-map visualization (right)

Music and Emotion

Baum et al. (2006) present a user study for implementing a classification system for music according to emotions. 200 music pieces out of the genres blues, classical, country, disco, hip hop, jazz, metal, pop, reggae and rock were therefore used. As emotional labels hostility, guilt, sadness, joviality, self-assurance, attentiveness, shyness, fatigue, serenity and surprise were selected. The study was arranged via a web-based questionnaire where each emotional category was represented by two adjectives that the participants could assign to a music piece. If more than five people agreed with the emotion, it was labeled by this category. In the next step rhythm patterns and a SOM were used to cluster the songs into a 2D map according to their sound similarity. Besides labeling the songs with its genre, the songs were also labeled with its assigned emotions to recognize relationships between them.

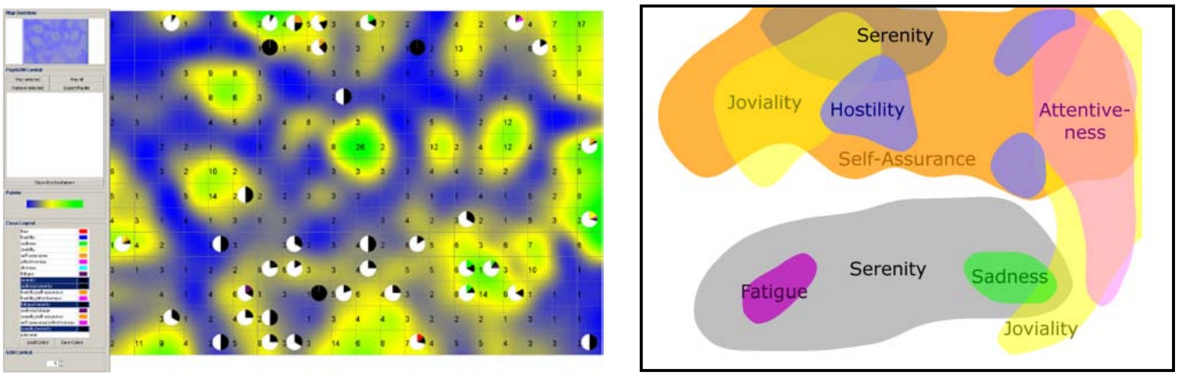


Figure-10: Distribution of serenity-labeled songs (left),
a manual constructed sketch of emotional categories (right)

In Figure-10 (left) the songs assigned to the class serenity are marked with circles for a visual comparability of genre and emotion agreement. Out of this user study a sketch of clusters of emotions was (manually) constructed as depicted in Figure-10 (right) to give an overview where songs, evoking specific emotions, can be generally localized.

Emotional Photo Slideshow

Chen et al. (2008) combine auditory and visual media to create an emotion-based music player. The player performs a slideshow of photos according to the auditory structure of a song. A subset of photos was manually annotated with the emotional categories sublime, sad, touching, easy, light, happy, exciting, and grant. Then visual features of the labeled photos were computed and a Bayesian classifier was used to automatically label 2.000 additional photos. A song is divided into segments, based on music emotion detection techniques an emotional label for each segment is extracted. A photo holding the same emotional category as a segment of the song is then assigned to it. Additionally the timbre of the music and the color information of images are taken into account to further harmonizing as well as rhythmical features of the song to switch the photos accordingly.

Music Icons

The Music Icon tool (Kolhoff et al., 2008) builds up special icons for songs to visually represent the internal structure of a song. Figure-11 (left) shows the layout of those icons: they are defined by a bloom-like shape, its form is automatically built up by specific parameters defining its colors and shape. First the user has to select icons for a set of prototype songs regarding what kind of music he best identifies with a particular type of icon. Then acoustic features are calculated from this prototype songs and a neural network is trained to suit the user’s preferences. After training, the neural network receives features of novel songs and maps them to an appropriate icon style. The software is available as a plug-in for the Microsoft Windows Explorer and for small devices as shown in Figure-11 (mid).

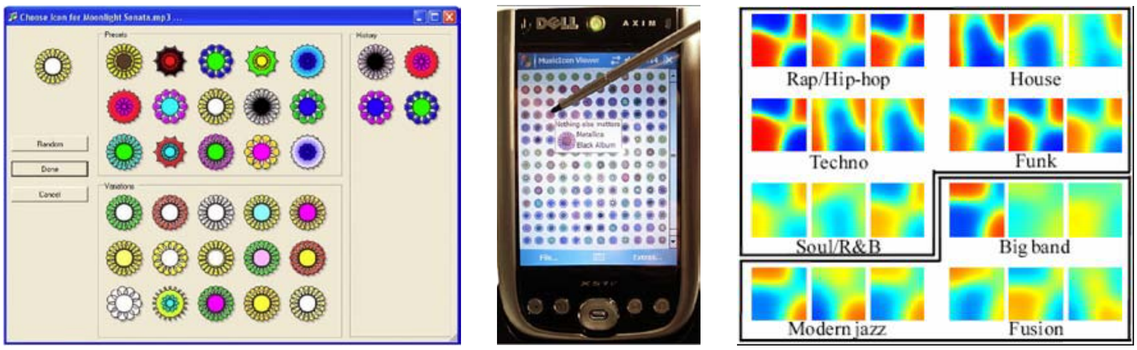


Figure-11: *Music Icons* (left & mid), the *Music Thumbnailer* (right)

Music Thumbnailer

The *Music Thumbnailer* (Yoshii et al., 2008) constructs appropriate thumbnails for music pieces to help users to guess the aural characteristics of a song without listening to it. A test music library is used containing 100 songs of 10 main- and 33 sub-genres. The implementation computes acoustic features of the audio files and uses a mapping model to transform the feature vector onto a 2D image which reflects the internal structure of the songs. Figure-11 (right) depicts several thumbnail appearances according to different genres.

Liveplasma

Liveplasma (Vavrille, 2005) is an online visualization tool to search and browse for music and movies. The user has to type in the name of an artist or a movie and an artist or movie similarity map respectively is build up accordingly. Figure-12 (left) shows a screenshot of the application[7], each artist is represented by a plasma cell. The queried artist is marked with an orange circle and surrounded by a limited number of similar artists. Clicking on a plasma cell of another artist builds up the map anew by adding or removing artist cells. In contrast to content-based audio analysis *Liveplasma* uses the `amazon.com` API for the recommendation of similar artists or movies but has no playback option.

Figure-12: *Liveplasma* (left), the *Shape of Sound* (right)

The Shape of Song

The *Shape of Song* (Wattenberg, 2001) is a song visualization tool which represents a musical work through translucent arcs. The result is an image which consists of several arcs reflecting the internal structure of a song; each arc connects two similar passages of a song. As an input file the MIDI format is used since it contains the coding of the notes for each instrument. Figure-12 (right) shows the resulting visualization of Madonna's *Living on a Prayer*.

The represented MIR and visualization tools in this chapter represent only a selection of a broad range of different approaches in organizing, exploring and visualizing music pieces and their features. Besides this selection, Isaacson (2005) gives a good survey of implemented music visualization and notation tools. Typke et al. (2005) outlines a more technical review of matching algorithms for both, audio and symbolic music formats, as well as existing MIR systems. Furthermore the *VisualComplexity.com* project website[8] serves as a good inspiration source for the visualization of music information. In the next chapter virtual environments are reviewed – several application areas are presented as well as selected tools for creating 3D worlds.

[7] Homepage of Liveplasma: http://www.liveplasma.com/
[8] Homepage of the Visual Complexity Project: www.visualcomplexity.com

2.3 Virtual Environments

The virtual dimension has captured us nowadays. Through the progress within computer technologies in the last years new possibilities established in creating Virtual Realities (VR). Based on workstations, immersive rooms, Head Mounted Displays (HMD), PC's, large screen systems and virtual tables there is a broad range in configuring VR systems.

> *VR is about using computers to create images of 3D scenes with which one can navigate and interact. (Vince, 1998, p.4)*

HMD's allow users to "see-through" a display to immerse and perceive a new - computer generated – world. This type of VR is also referred as an *immersive* VR since it is the most direct experience where the user becomes an active part of the VR, and may navigate and interact within this world. *Non-immersive* VR systems are based on a portal or a window (for example a standard computer monitor) where the virtual environment is viewed through - examples are CAD[9] and VRML applications or 3D computer games (Vince, 1998).

In Figure-13 the *Reality-Virtuality Continuum* defined by Milgram et al. (1994) is presented. The continuum spans an area ranging from purely real environments to purely virtual environments where all levels in between are referred as to be a mixed reality.

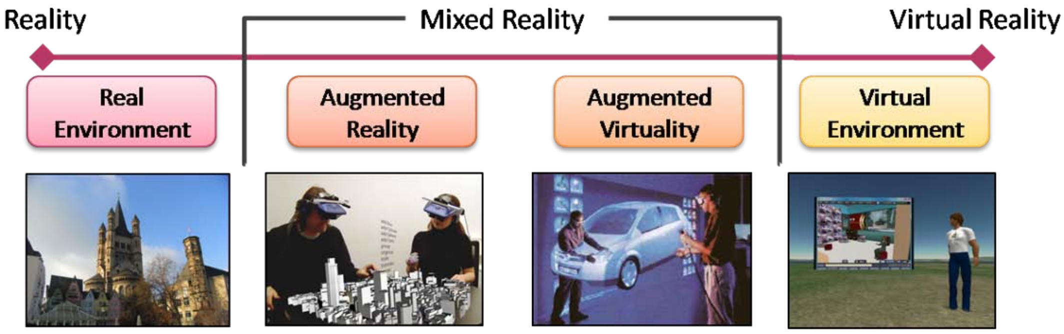


Figure-13: Reality-Virtuality Continuum – adapted from Milgram et al. (1994)

Moving from the left to the right on the line means an increase of virtuality or a decrease of reality respectively. A real environment is defined as an observation when looking at the real world (directly as a person, either through a window or a camera display) and clearly follows the laws of physics. On the other hand pure virtual environments consist only of virtual objects and are for example monitor-based computer graphic simulations or 3D worlds. Within augmented realities, virtual objects are add-

[9] CAD - **C**omputer **A**ided **D**esign

ed to a real scene - for example through HMD's. In contrast within augmented virtually physical objects are integrated to the scene – for example through stereoscopic projection techniques.

2.3.1 Application Areas

In this chapter selected application areas of virtual environments are presented. They can be designed for a single (individual) user or for multiple (social) users. On the one side the focus on a virtual environment for a single user may lay on the design and organization of the 3D world and its interaction possibilities with the environment. And on the other side the main goal for multiple user environments are the communication and the interaction with other users in such a social network. Such tasks could be studying documents via teleconferencing, collaborative work with text, videos, and graphics, interaction with integrated applications or simply social interaction (Chen, 1999). This chapter presents virtual environments out of the following areas:

- Virtual Online Communities
- Virtual Museums
- Cultural Heritage
- Scientific Visualization
- Virtual Workspaces

Virtual Online Communities

There exist numerous virtual 3D world installations spread over the World Wide Web (WWW). Users can participate at existing worlds, communicate with other users in form of avatars and can also create new worlds.

In 1999 Philip Rosedale had the idea to create a new form of shared experience within the WWW which resulted in the *Second Life (SL)* world (Linden Research Inc, 2010). *Second Life* is online since 2003 and offers tools for business, educators or nonprofit organizations to develop a virtual presence whereas the 3D locations are designed by its residents themselves. Millions of people from all over the world are registered within the *Second Life* community and for an on-going presence in *Second Life* users can buy land; this may range from a tiny parcel to an entire island. Apfelthaler (2009) outlines in detail several application areas of *Second Life* as arts and cultures, historical and modern places and cities and virtual museums and universities.

Besides *Second Life* there are a great number of other 3D online worlds; *Active Worlds*[10], *There*[11] or *Habbo Hotel*[12] are just a few platforms to mention. Furthermore the website *Virtual Worlds Review*[13] gives a good summary of existing virtual online worlds.

The users are represented inside the virtual worlds as avatars and can select avatar specific features like name, clothes, hair and skin color and much more. Virtual scenes may be built over millions of square kilometers of virtual territory and designed as modern-day cities, bars and restaurants, beaches, landscapes, undersea and extraterrestrial scenes; all based on either real-world exemplars or imaginary scenes. Virtual worlds for communities often provide features like voice or text chat. Environmental sound effects, animations of waterfalls, rain, the sun-movement, day-night changing may all be integrated parts. On entering such virtual worlds one recognizes that there are no restrictions in the design and embellishment of virtual environments.

Figure-14: Music island[14] in *Second Life* (left), a virtual world in *There*[11] (right)

Figure-14 (left) shows a screenshot of a virtual concert on the *Music Island*[14] in *Second Life*. Figure-14 (right) shows a possible scenario within the online community *There*; logged in users are meeting with others to virtually play cards while performing text chat.

Virtual Museums

Virtual museums are being modeled with increasing popularity since they have several advantages compared to real museums. They are able to exhibit objects which were not possible to present in real museums. This can be for instance ancient objects which no longer exist or are just partially preserved. Or exhibits which are not suited for a real museum due to lack of space, need for special handling or

---

[10] Homepage of Active Worlds: www.activeworlds.com
[11] Homepage of There: www.there.com
[12] Homepage of Habbo Hotel: www.habbo.de
[13] Homepage of Virtual Worlds Review: www.virtualworldsreview.com
[14] Music Island in SL: http://world.secondlife.com/place/5021f4a4-ff62-fcc0-0333-57b5bed75e04

not easy to visit or too large in size. Multimedia, 3D graphics and virtual reality are used to enhance the presentation for a more vivid and enjoyable experience.

Edutainment (education through entertainment) installations are established by giving the user the opportunity to interact and "play" with artifacts; for example rotate, move or decompose them, or even reassemble exhibits, which is obviously not possible in most real museums (Tsichritzis et al., 1991).

In *Second Life* numerous museums already present their exhibits. One example is the *Dresden Gallery*[15] which was the first true to scale and completely implemented 3D museum in *Second Life*. It is online since 2007 and exhibits important artworks like Raphael's *Sistine Madonna* or Giorgione's *Sleeping Venus*. Visitors can explore the virtual museum 24 hours a day and 7 days a week; Figure-15 (left) shows a screenshot of the first floor of the gallery. Other examples in *Second Life* are *The Second Louvre*[16], the *Second Life Historical Museum*[17] and the *International Spaceflight Museum*[18].

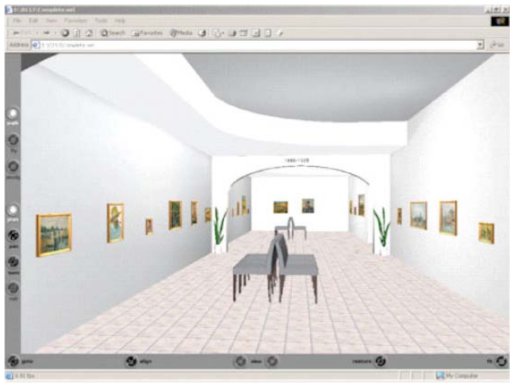



Figure-15: The Dresden Gallery[15] in *Second Life* (left), customized Van-Gogh gallery using VRML/X3D (right)

CiVeDi (Mazzoleni et al., 2004) is a virtual exhibition environment for accessing multimedia content that is built up dynamically. The user can select the artifacts to be shown up, and then a customized museum is constructed. The size and the appearance of the museum (i.e. ancient or modern) are incorporated by the number of artifacts and their characteristics. For the implementation Java was used and X3D for specifying dynamic templates of museum parts. Figure-15 (right) shows such a customized gallery using only a specific number of art works of Vincent Van-Gogh.

The *Virtual Museums Project* (Lepouras et al., 2004) focuses on the comparison of virtual reality technologies for implementing virtual museums. Based on an already implemented version of a museum they re-implemented it using game-technologies. They concluded that for edutainment tasks

[15] Homepage of the Dresden Gallery: www.dresdengallery.com
[16] The Second Louvre Museum in SL: http://secondstuff.wordpress.com/2009/01/13/the-second-louvre-museum/
[17] The Historical Museum in SL: http://world.secondlife.com/place/a7177dcd-9542-d55d-7763-469d4791046a
[18] International Space Flight Museum in SLhttp://world.secondlife.com/place/4c428aeb-edb0-3a84-068d-7f9974d3ee4c

game-engines are a better solution than other technologies since they provide an expandable environment and a fundamental infrastructure in navigation and interaction.

Mateevitsi et al. (2008) present a virtual museum using the *Torque Game-Engine*. The museum consists of several artifacts, whereas a description appears when the user comes close to it. Metadata are used to provide search options by name or creator or to query similar exhibits. When the user selects an exhibit from the result list, an intelligent museum avatar guides the visitor to the requested object.

Reconstruction of Ancient Sites

The preservation of ancient cities, sculptures or scenarios is for instance archived by reconstructing them in a virtual environment. One institution is *The Foundation of the Hellenic World* (FHW) located in Greece, which is working on technologies for 3D modeling the cultural heritage of the ancient *Hellenic* world (Gaitatzes et al., 2001). The implementation of *The Ancient City of Miletus*, *The Temple of Zeus at Olympia*, Figure-16 (left), or an *Olympic Pottery Puzzle* are just a few project at the Foundation's *Cultural Center for Entertainment and Edutainment* where people can virtually interact with ancient objects. In addition computer graphics and projection-based virtual reality techniques are experimented with to perceive ancient places in a new way.

Figure-16 (right) shows the 3D reconstruction of the Mexican ruin Chichen-Itzá with its *Pyramid of Kukulcán* and the *Venus-Platform* which is online in *Second Life* since the year 2007[19]. It is just one of several historical places in *Second Life* like the *Ancient Rome*[20] or *Paris in 1900*[21].


Figure-16: The Temple of Zeus at Olympia (left), Chichen-Itzá in Mexico[19] in *Second Life* (right)

Furthermore the website of *3D Ancient Wonders*[22] exhibits 3D versions of ancient artifacts, relicts and architectures like *The Great Pyramids of Gizeh* or *Stonehenge*.

[19] Online Article about Chichen-Itza in SL: http://www.associatedcontent.com/article/274652/chichenitza
[20] Ancient Rome in SL: http://world.secondlife.com/place/d5d01046-9f8b-f7ec-814b-881e5b5827dd
[21] Paris in 1900 in SL: http://world.secondlife.com/place/44a67009-5675-b47f-04a4-058115a0a238
[22] 3D Ancient Wonders: http://www.3dancientwonders.com/

Scientific Visualizations

Virtual environments also serve as platforms for scientific information exploration. The aim is to give one or multiple users the opportunity to navigate through complex data structures in 3D spaces for a far better comprehension than a 2D visualization. Application areas range from 3D galaxy visualizations, geographical visualizations of WWW traffic, co-citation maps for an efficient analysis of publications, to website structure visualizations and site maps, to database explorations and Geographic Information System (GIS) visualizations (Chen, 1999).

Figure-17 (left) depicts a molecular visualization implemented in a virtual learning space. Figure-17 (right) depicts a screenshot of a 3D visualization of a communication network realized by the company 3DSolve - both applications are implemented through the open source software *Open Croquet* (Croquet Consortium, 2010) and can be found on its website.

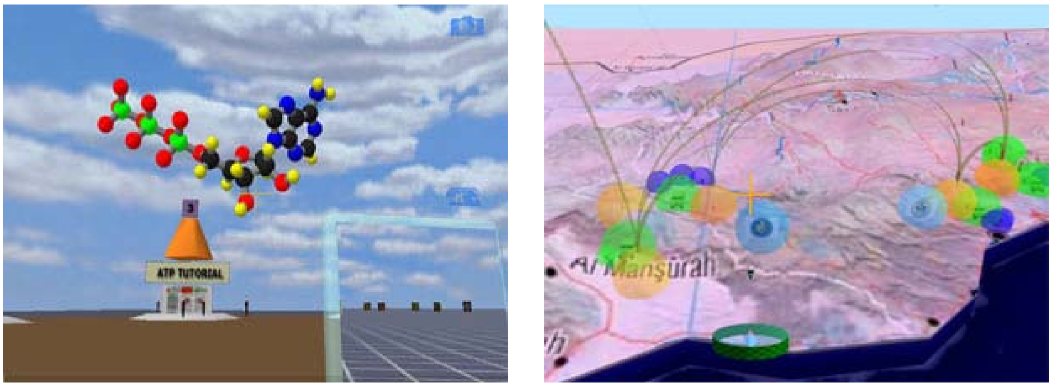


Figure-17: 3D Molecular Visualization (left), 3D Communication Networks (right) both using *Open Croquet*

Virtual Workspaces

Virtual workspaces (or media spaces) provide users with an audiovisual infrastructure to communicate with each other. Collaboration is a fundamental part of such systems and audio- or video conferencing systems, shared desktop applications or whiteboard tools are common technologies in supporting user's workflows. The environmental designs of such systems are for example virtual offices or conference rooms.

> *The ultimate goal of a media space is to help people to transcend the limits of the physical world and work harmoniously and productively. (Chen, 1999 p. 181)*

Figure-18 shows a screenshot of Sun's virtual workplace MPK20[23]. Sun's physical Metro Park (MPK) campus consists of 19 buildings numbered from MPK1 to MPK19 and they decided to build their next

[23] Homepage of SUN`s MPK20: http://research.sun.com/projects/mc/mpk20.html

building MPK20 in the virtual world. MPK20 is a virtual environment for business collaboration and consists of a team room and several private rooms. Employees of SUN can accomplish their real work, share documents and meet with colleagues using voice communication. The implementation of MPK20 is based on Sun's project *Wonderland* toolkit (cp. next chapter).

2.3.2 Tools

In this chapter tools for implementing virtual environments (or 3D worlds) are presented and shortly described in terms of their fundamental technology and range of functionalities.

Project Wonderland

Project Wonderland (Sun Microsystems, Inc., 2010) is an open source Software Development Kit (SDK) for building 3D interactive virtual environments. It is completely written in Java and licensed under the GNU General Public License. *Project Wonderland* is aimed at the creation of collaborative 3D worlds where users can communicate, share live desktop applications and documents and conduct real business as communicate with customers, partners and employees.

Figure-18: Virtual Workspace using Sun's *Project Wonderland*.

Project Wonderland can either be downloaded as a binary distribution or be built from source code. After the installation process, an instance of the *Wonderland* server is created which includes a web server component and several default Wonderland worlds. A *Wonderland* world can be launched by providing the users with a link on a webpage where the client is then launched via Java web start.

Wonderland supports "in-world" 2D and 3D applications. A 2D application can be a multi-user white-board, a slide-show-view for PDF documents, sticky notes or a web cam viewer. The *Wonderland* server can also be configured to run any X11 applications like Firefox, OpenOffice, or NetBeans. These applications can only be controlled by a single user, but every other user with permission can view and watch the documents being updated. 3D applications are for example an audio recorder the "Cone of Silent" or the brainstorming pads.

Developers and graphic artists can extend the functionality of Wonderland to create entirely new worlds and add new features to existing worlds, like new behaviors for objects and avatars. Server and client can be extended by creating so called modules (which are similar to plug-ins in other systems). Such modules can be interactive tools, like a 3D molecule simulation, a megaphone application or a Java Swing application. Wonderland supports 3D artwork import in the COLLADA[24] data format which can be exported from graphics tools like Maya, 3D Studio Max or Blender. The current version of *Project Wonderland* is 0.5 (as of 05.02.2010) and the system requirements for running it are a modern PC's with 1.5 GHz+, 1 GB RAM and hardware-accelerated OpenGL drivers installed. In addition a graphics card which gear towards 3D games and a video memory of 256 MB is required.

Open Croquet and Cobalt

The *Croquet* SDK (Croquet Consortium, 2010) is an open source SDK to create collaborative multi-user applications on multiple operating systems and devices. The rendering framework of *Croquet* is built on top of OpenGL. Croquet is an extension of *Squeak*, a modern variant of *Smalltalk*. Within the *Croquet* SDK there is no distinction between the user environment and the development environment and worlds can be modified or updated while the system is live and running.

Croquet supports communication, collaboration, resource sharing, and synchronous computation between multiple users. The user interface is a collection of objects that can be replaced or enhanced. Objects can be 2D or 3D content, text annotation tools, in-world web-browsers or other 3D worlds. Croquet follows a peer-to-peer network architecture called *TeaTime* which is the basis for communication and synchronization of the so called *TeaObjects*. Objects can be selected and manipulated. Messages sent to a *TeaObject* are redirected to copies on other user's machines which participate in the network. Through this architecture it is possible to coordinate the activities of users within virtual worlds without maintaining central server resources. Any user has the ability to create and modify a "home world" and create links to any other such world. It is possible to view from one world into another and the user can enter a new world by walking through its portal. *Croquet* was released as a SDK in 2007 and since then active development of the *Croquet* system is taking place in the *Open Cobalt* project[25].

Torque Game Engine

The *Torque Game Engine* (TGE) (GarageGames, 2010) is a commercial SDK implemented in C++. It follows a server-client architecture to support multi-user gaming over the network and live text chat.

[24] COLLADA (**COLLA**borative **D**esign **A**ctivity), an XML based exchange format between 3D programs www.collada.org
[25] Homepage of the Open Cobalt Project: www.duke.edu/~julian/Cobalt

Torque 3D editors are graphical user interfaces to design a game without modifying the source code. The editors provide functionalities to create and modify game levels, terrains, game objects, environmental effects, lighting and animations. Further the main menu, splash and loading screens can be configured. The game logic is controlled through *TorqueScript* a script language similar to C++.

In terms of 3D artwork the TGE supports the DTS format (used for large, static models like buildings) and the DSQ format (used for small, dynamic and animated objects, like characters or vehicles). Common 3D modeling tools like Maya or 3D Studio Max support the exportation to the DTS format. In addition TGE has the ability to import COLLADA files.

As a rendering system GFX is used which is an abstract graphics layer designed to reside above graphics APIs such as Direct3D and OpenGL. The SFX system is used by TGE to support audio playback. It supports the WAV and Ogg/Vorbis format, positional (3D) sound and non-positional (2D) sound whereas 3D sound is only supported for a single channel (mono) audio signal. Through SFX audio can be played back either buffered or streamed. In addition the SFX system supports several sound API´s like OpenAL[26] or FMOD. System recommendations for running TGE are a modern PC having 2 GHz+, 2 MB RAM and a DirectX compatible graphics card with 1 GB video RAM.

VRML/X3D

VRML (Virtual Reality Modeling Language) development began in 1994 by Mark Pesce and Tony Parisi with the aim to create a markup language to design 3D scenes on the web. In 1997 the specification of VRML 2.0 (or VRML 97) was approved as the International Standard ISO 14772. VRML scenes are displayed in real-time via an adequate web browser or a plug-in for a standard browser. A popular VRML interpreter is the *Cortona Player* by Parallelgraphics which is suitable for Mozilla, Firefox or Internet Explorer or FreeWrl for Linux.

A VRML file is readable text file (`.wrl` extension) where a simple text editor can be used for scripting. A 3D world described with VRML follows the scene graph model that uses a directed and acyclic tree to collect all aspects of a 3D scene (i.e. 3D models, light, sounds, videos, internal relationships, relative locations, appearances, animations and events). Furthermore 3D objects or complete scenes can be animated and interacted with. Objects of a scene can also represent hyperlinks to other VRML worlds the user can be teleported into by clicking on them. So called script-nodes can contain scripts or programs to control complex behaviors or manipulations of objects which were not possible only using VRML. Supported languages therefore are Java, JavaScript, C/C++ or Perl. Scripts and pro-

[26] Homepage of OpenAL: http://connect.creativelabs.com/openal/default.aspx

grams are also used to manage network communications over the internet and establish multi-user VRML worlds (Hase, 1997).

Now VRML has entered its third generation as X3D (Extensible 3D Graphics), was approved by ISO as ISO/IEC 19775 in 2004 and is guided by the Web3D Consortium (Web3D Consortium, 2010). Within the X3D standard the VRML scene graph was mapped into XML to become much more compatible with web-based technologies. The specifications of X3D are a detailed set of technical documents that define the geometry and behavior capabilities of Classic VRML using the tag set of XML (Brutzman et al., 2007).

Java3D

The Java3D API was developed by Sun Microsystems (Sun Microsystems, Inc., 2010) and its recent version is 1.5.2. (as of 10.02.2010). Since it was chosen for the implementation of MeXX it is described in more detail in Chapter 3.2.

Two further tools that establish virtual worlds are *Open Simulator*[27] and *Open Scene Graph*[28]. *Open Simulator* is a 3D application server for building up virtual worlds which then can be accessed through a variety of clients like the *Second Life* viewer. With *Open Simulator* it is possible to simulate a virtual environment similar to *Second Life* but without the need of buying land. *Open Scene Graph* is an open source platform independent 3D graphics toolkit. Written in standard C++ and OpenGL it is used for visual simulation, games, virtual reality, scientific visualization and modeling.

2.4 Music-Based Games

Music-based games are games which are designed around an auditory experience. Music plays the central role and a player's interactions are oriented around one or more music pieces. Music-based games are grouped into the following categories (Pichlmair et al., 2007):

- Rhythm Action Games
- Play as Performance Games
- Free-form Play Games

Rhythm Action Games

So called *Rhythm Action Games* challenge a player's sense of rhythm. The aim of such games is that the player presses a button at the right time to collect points in terms of being on the beat and accuracy. As progress is made by the player the game successively increases the speed and difficulty of the

[27] Homepage of Open Simulator: www.opensimulator.org
[28] Homepage of Open Scene Graph: www.openscenegraph.org

rhythms. *PaRappa The Rapper*[29] by NanaOn-Sha is one of the first music-rhythm games released in 1996. The player has to repeat an instructors rap in the same rhythm. *Guitar Hero* by Harmonix (from 2005 to 2007) respectively Neversoft[30] (since 2007) challenges one or more players to rhythmical correctly replay a song by using a special guitar-controller. Figure-19 (left) depicts the game layout of *Guitar Hero* for three users and the guitar controller on the right.

Within *Vib-Ribbon*[31]*,* released in 1999 by NanaOn-Sha, a song is analyzed and mapped as an obstacle on the screen. In Figure-19 (right) a screenshot of *Vib-Ribbon* is shown. The player has to go around the obstacles by pushing the right button. A special feature of *Vib-Ribbon* is that it loads the audio totally into the memory to calculate the internal structure of a song in real-time. Thus any customized songs can be loaded into the game. Due to that sound handling more simple graphics are used by this game.

Figure-19: Guitar Hero[30] (left), Vib-Ribbon[31] (right)

Play as Performance Games

Play as Performance Games are games where players are prompted to actively perform “on stage”. All these games offer no possibility of expression on the virtual part (the displayed graphics of the game) but a wide range of expressive possibilities in the real world. These music-based games are popular party games and games for “non-gamers”.

Examples are the game *Sing Star*[32], a karaoke game from Sony Computer Entertainment where players have to sing a song and being challenged by their accuracy in terms of rhythm and pitch. In *Dance Revolution*[33] by Konami a player is instructed to jump between different areas on a mat controller ac-

---

[29] Homepage of PaRappa The Rapper: http://us.playstation.com/parappatherapper/
[30] Homepage of Neversoft, Guitar Hero: http://www.neversoft.com/
[31] Homepage of Vib-Ribbon: http://www.vibribbon.com/
[32] Homepage of Sing Star: http://www.singstargame.com/
[33] Homepage of Konami, Dance Dance Revolution: http://de.games.konami-europe.com/game.do?idGame=229

cording to the beat of a song. Also *Guitar Hero* falls in that game category since the player gets extra points if he hoicks the guitar at the right moment.

Free-Form Play Games

Free-form Play Games are more instruments than games, the game aspects are postponed in favor of the instrument aspects. An example is *Electroplancton*[34] by Nintendo where the player can produce music by stimulating aquatic creatures to produce noise, melodies, techno or industrial beats. Figure-20 (left) shows a screenshot of *Electroplancton*. The player can experiment with sounds, distort them and create an individual sound collages and own melodies. *Sim Tunes*[35] by Maxis is another example game which is mainly made for children to create "music-pictures". Figure-20 (right) shows the game layout, the player creates his own music melodies by drawing dots on the board that represent musical notes. *Bugz* represent different musical instruments, sounds, effects or singing voices (sound features are indicated again through colors) and can be referenced by the notes to create a sound collage. The speed and direction of a sequence of notes can also be modified by the player.

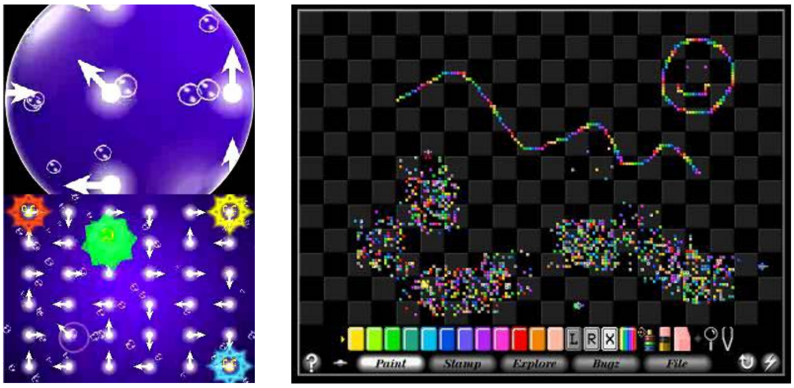


Figure-20: Electroplancton[34] (left), Sim Tunes[35] (right)

Kayali (2008) outlines the history of music-based games and the experience the users receive during playing. Furthermore he examines the technical foundations and design principles when creating such a game. Marschner (2008) presents the history of music in games in detail. In addition he outlines how to extend the *Torque Game Engine* in terms of game pad controllers and appropriate sound libraries to serve as a base for music-based games.

Since mainstream computer games can in general not be played by people with visual restrictions, the *TiM Project* (Friberg et al., 2004) is working on music-based games which can be used by both, people with visual impairment and people without any restrictions. They focus on new computer en-

[34] Homepage of Nintendo, Electroplankton: http://www.nintendo.de/NOE/de_DE/games/nds/electroplankton.html
[35] Homepage of Sim Tunes: http://www.mobygames.com/game/simtunes

tertainment games that feature complete auditory interfaces. An additional source for music-based games is the website of *AudioGames.net*[36] which is an informative online platform.

2.5 Music Environment Systems

In this chapter selected music environment systems implemented in either a virtual or a hybrid world are presented as well as real-world music installations. They are described in terms of their fundamental technology and intention of usage.

Second Life

Within the *Second Life* world, the *Grimes Megastore*[37] exhibits music equipment like pianos, guitars, DJ & club equipment, lights and music furniture. When navigating through the store, the user is provided with descriptive information about the instruments, and can "try-out" pianos by clicking on a music sheet located at them. Figure-21 (left) shows a second store, the *Agenda Megastore*[38]. When entering this place several instruments are exhibited that emit musical patterns of their sound characteristics.

Figure-21: *Agenda Mega Store*[38] in *Second Life* (left), the Bara Joson concert[39] in *Second Life* (right)

A "live" concert was performed in *Second Life* by the singer Bara Joson at the virtual presence of the *Kulturbiergarten* located at the inner city of Munich[39]. At the same time the virtual concert was transferred on a live screen at the real Kulturbiergarten as an experiment; Figure-21 (right) shows a screenshot of this virtual concert. *Second Life* consists additionally of numerous music and dance clubs and bars as well as theatres, ballrooms, music stores and even more places which all deal with music and fun to explore.

[36] Homepage of AudioGames.net: http://www.audiogames.net/
[37] Grimes Megastore in SL: http://world.secondlife.com/place/d583efbb-c978-2dd3-04ae-5d3e68de8670
[38] Agenda Megastore in SL: http://world.secondlife.com/place/68d23a56-1fed-3ea9-ea47-1c0429543d50
[39] Homepage of Munich in SL: http://www.echt-muenchen.de/blog/2009/08/bara-jonson-live-im-kulturbiergarten.html

The Audio Square

The *Audio Square* (Genswaider, 2008) is a multi-user application aimed at the spatial perception of music. Figure-22 (left) shows a top view screenshot of the system. The sound is emitted from 3D furniture objects composed of a table, chair, playlist and a loudspeaker. The users are represented by avatars and can navigate through the Audio Square to listen to different songs located in two showrooms. Within the *SOM Showroom* music pieces are mapped by their similarity through the SOM approach into the building as depicted in Figure-22 (right). The *Manual Showroom* is built up through a predefined directory structure of musical works. Buildings for different genres are created and contain music pieces which belong to that particular genre.

The *Audio Square* is implemented using the *Torque Game Engine*; the songs are received via audio streams over the internet from the *Magnatune*[40] music library. The participants are able to communicate via a live chat system with other users during examining the Audio Square. The *Media Square* (Dittenbach et al., 2007) is an extension of the *Audio Square* where multimedia documents including images, posters or videos are represented within separate scientific library buildings.

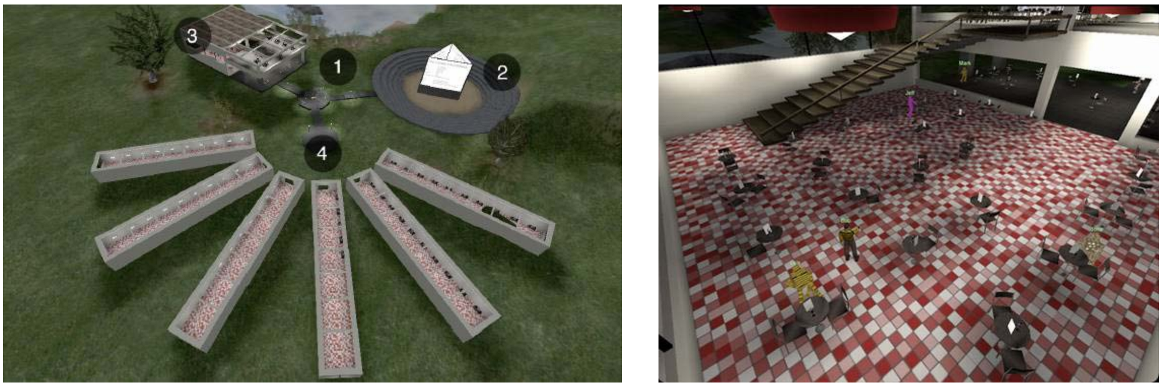


Figure-22: Overview about the Audio Square (left), the SOM Showroom (right)

nepTune

Within *nepTune* (Knees et al., 2007) a music repository is mapped onto a virtual landscape which follows the *Islands of Music* approach (Pampalk, 2003). Songs that belong to specific genre form a 3D island where the height indicates the number of songs that belong to it. By navigating through the landscape the user perceives the nearest songs which surround him. Besides listening to the audio, also contextual information is provided and the user can switch between four display modes: the first mode depicts the track and artist names as shown in Figure-23 (left), the second mode hides all information and therefore allow the user to concentrate on the pure audio experience, within the third mode addi-

[40] Homepage of the Magnatune Music Library: www.magnatune.com

tional text descriptions show up and the fourth mode displays images retrieved automatically from the internet (Figure-23, right).

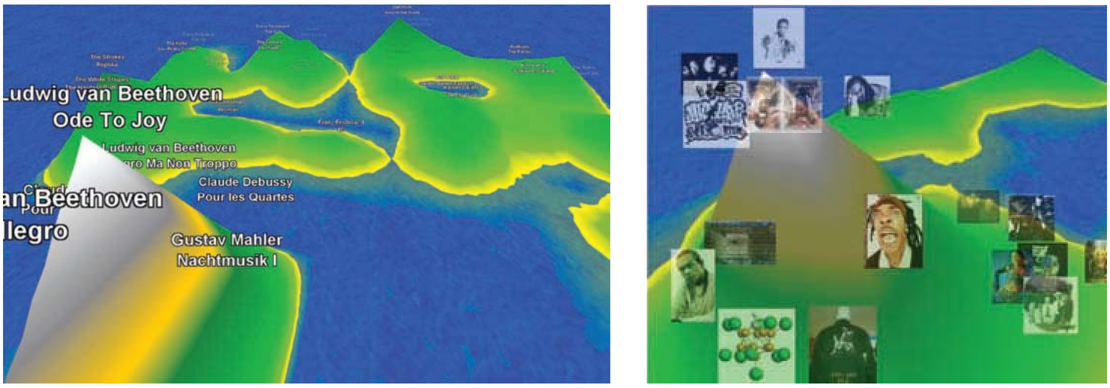


Figure-23: The *nepTune* user interface

In addition to common computer monitors *nepTune* may be used in a public exhibition space where visitors can connect and explore their own music collections. The system processes and analyses the audio files and builds up the 3D landscape which is then projected on a screen. A 5.1 surround audio system is used for a suitable 3D perception of the music and a game controller serves as a navigation device.

Search Inside The Music

Search Inside The Music (Lamere et al., 2007) represents music pieces through spheres in a 3D space. The music pieces are arranged by their similarity where models based on the *MARSYAS*[41] software are used. An automatic play list is generated when the user selects a start and an end song.

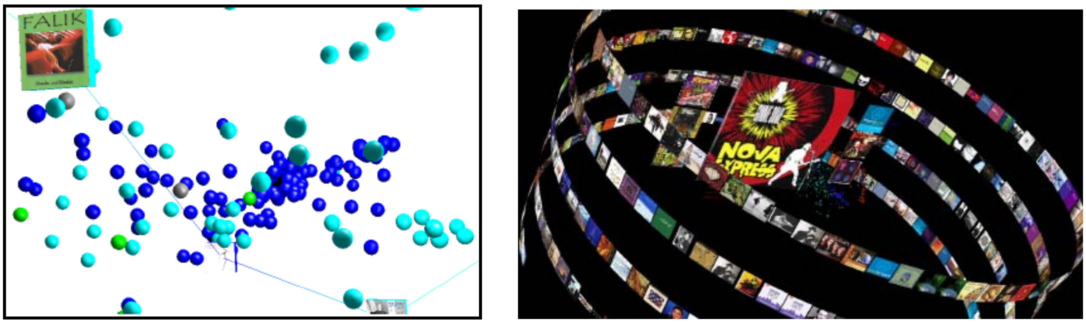


Figure-24: Play list generation in *Search Inside the Music* (left), album grid visualization (right)

Figure-24 (left) shows a playlist path within a 3D space – the playlist's start and end songs are marked with their related album covers. Greater spheres indicate favored songs by the listener played more frequently. The system also provides functions for a geometrical visualization of the album artwork.

[41] MARSYAS (**M**usic **A**nalysis, **R**etrieval and **SY**nthesis for **A**udio **S**ignals) http://marsyas.info/

Shapes like rings, boxes, ellipses, planar grids or other structures can be selected to arrange album covers on them. Figure-24 (right) shows a visualization where one album has become the central element and is surrounded with similar albums organized on a spiral.

The Globe of Music

The *Globe of Music* (Leitich et al., 2007) is a spherical visualization tool for music libraries and depicted in Figure-25 (left). Content-based features are extracted from the audio files and the *GeoSOM* - a spherical SOM – is used to cluster the audio files into genres. The music pieces are mapped into a *World Wind* system which is developed by the NASA and provides users with functionalities to explore spherical objects like planets. The sphere consists of icons representing music pieces textured by the corresponding album's artwork, the music pieces are interlinked to others that are similar.

PODIUM

The *PODIUM* (POstech Distributed virtUal Music environment) application (Jung et al., 2000) is a collaborative 3D environment. The aim of this system is that users can virtually play music together. The participants operate on virtual instruments using the computer keyboard, the mouse or other devices to generate notes. The right picture in Figure-25 shows a screenshot of *PODIUM* with two users which virtually play the piano together. The information is then transmitted through an IP-multicasting network among the users. Predefined animations of avatars according to their role in this environment are realized whereas real world interaction behaviors are reproduced. One role could be a musician playing an instrument, a conductor or an audience performing applause.

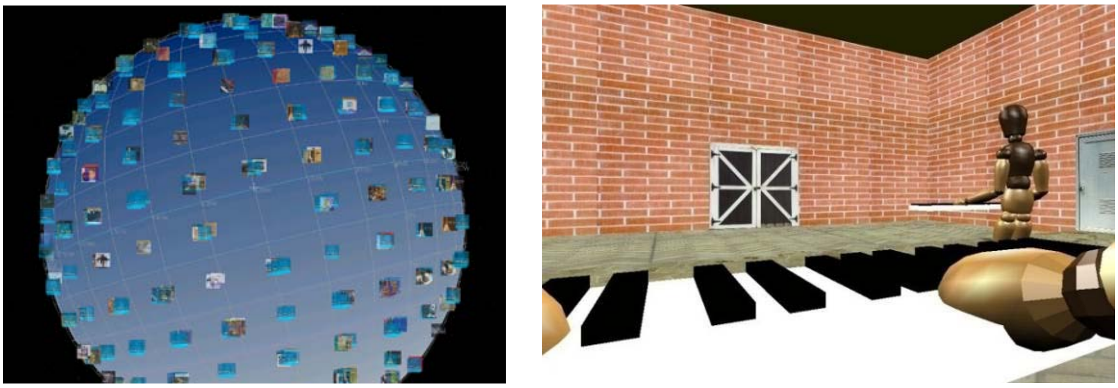


Figure-25: The Globe of Music (left), PODIUM (right)

Jam Tomorrow

Jam Tomorrow (Thalmann, 2006) aims at collaborative music generation. The world consists of several so called music editors which are the graphical music production interfaces the users are working with. Through these music editors one or more users can collaboratively add, modify and play different types of tracks. *Jam Tomorrow* was implemented using *Open Croquet* for building up the 3D environment and OpenAL to produce 3D sounds.

3D Concert Simulation

Research on an automatic generation of a realistic concert environment is done by Yilmaz et al. (2006). It is mainly focused on realistic graphical rendering of animations of crowds and thus on their behavior in specific concert situations like moving or raising hands, clubbing with the drum beats or jumping in the same rhythm with others. The environment consists of three basic elements: the crowd, the concert arena and the performer. The system requires an audio file as input; relevant features including genre, mood, beats per minute and rhythm are extracted. According to these features automatically an adequate concert scenario is built up by determining specific audience and band attitudes that match the style of the used song. One possible concert scenario could be a rock-concert with a groovy band performing and a young audience jumping to the rhythm of the beat as depicted in Figure-26 (left).

The musicCave

The *musicCave* (Frank et al., 2008) is an immersive music environment making use of virtual reality techniques. Figure-26 (right) displays a typical application scenario; within this installation 120 songs are clustered on a 6x6 SOM according to their rhythmic similarities using the *PlaySOM* software (cp. Chapter 2.2.2). The music map is projected on a 4-wall *Cave Automatic Virtual Environment* (CAVE) where each node of the SOM is represented by a red hemisphere – the bigger the hemisphere the more songs are located within this area. Through 3D goggles the user perceives the projection in stereoscopic views and can move around in the 2.5 x 2.5 meter sized CAVE by either walking or using a joystick. Each node plays a song and the user perceives a single song or a mixture of several songs when exploring the *musicCave*; the loudness and sound direction depend on the distance and angle of the user's position to the nodes.

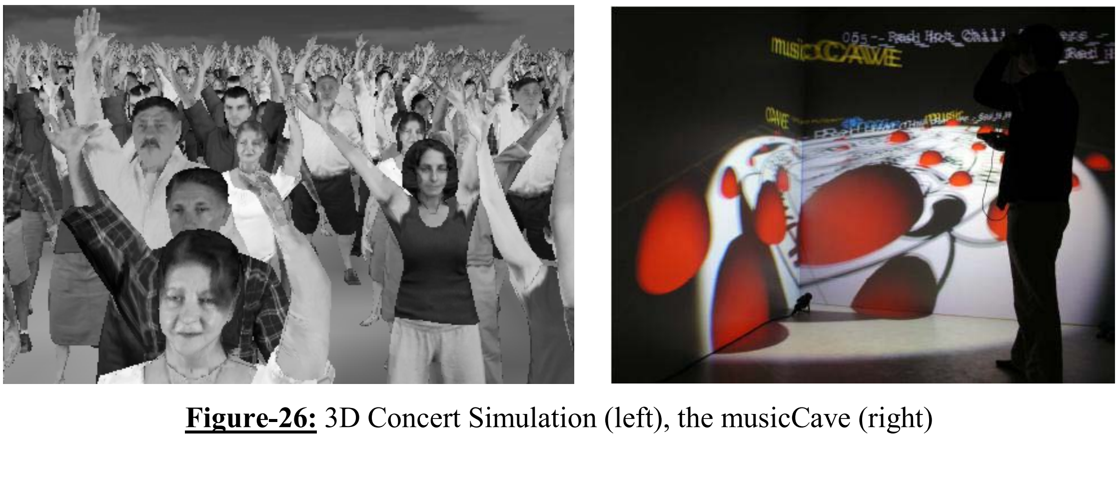


Figure-26: 3D Concert Simulation (left), the musicCave (right)

The MUSICTable

The *MUSICTable* (Stavness et al., 2005) is a multi-user music selection system installed in a social environment. Its main aim is to prevent the so called "separate party syndrome" where a subgroup of people tends to separate from the others to dominate the music selection. The installation displays songs on a music map classified into 50 categories according to their sound and genre similarities. The names of song and artists are kept away from the users to exclude a specific song selection. Figure-27 (left) shows a picture of the installation: the music map is displayed on a tabletop which is surrounded by eight buttons arranged at compass directions. The *MUSICTable* focuses on a collaborative music selection where each user input is taken into account for the next song selection. Several users at the same time can move a selection cursor in that eight directions by pushing the according buttons to collaboratively determine the location of the next song. If the current song ends a randomly selected song from the nearest area according to the cursors position is being played next.

Klangwiese

Baumann (2005) presents a physical music installation called *Klangwiese*. The aim of this project was to realize soundscapes as a public exhibition. Therefore the *Fruchthalle Kaiserslautern* was transformed into a *Meadow of Music* by covering it with real pre-grown lawn and 60 music flowers; Figure-27 (right) depicts this scenario. The music flowers consist of a speaker and an mp3-player that played five songs in an endless loop. The *Klangwiese* resulted in a huge analog interface where each visitor was able to browse for music in its own style, tempo and taste.

Figure-27: The *MUSICTable* (left), the *Klangwiese* (right)

2.6 Summary

Chapter 2 gave an overview about the research fields Music Information Retrieval (MIR), virtual environments, music-based games and music environment systems. In section 2.2.1 the fundamental features of music were listed like the tempo, the pitch or the harmony. It was outlined that a music piece can be represented by symbols or its audio waves but furthermore can be described using metadata. All three possibilities serve as an informative basis of music pieces and sounds and provide a great research basis in working and analyzing them. In the following section 2.2.2 MIR systems which provide novel graphical user interfaces to browse music collections were presented. Afterwards section 2.3 sketched the Reality-Virtuality Continuum by Milgram et al. (1994) and several application areas of virtual environments were presented including virtual online worlds, virtual museums and virtual workspaces. In addition 3D worlds are used to reconstruct ancient sites and for scientific visualization issues. Finally tools for implementing a virtual environment were presented. Section 2.4 dealt with music-based games. In such games music or sound is the central element to challenge the player's sense of music. They are categorized in rhythm action games, performance games and free-form play games. In the recent section 2.5 virtual and hybrid music environments as well as music installations in the real world were examined. In the following chapter the concept of MeXX is presented including its purposes and several application scenarios.

3 Development of MeXX

In this chapter the underlying concept of MeXX and the following implementation are outlined. In Chapter 3.1 the process in developing the final concept is presented and the underlying database structure designed for MeXX is sketched. Chapter 3.2 claims the Java3D implementation.

3.1 Concept Formation

This chapter describes the concept of MeXX and its backgrounds that led to the decision to develop it. The basic idea was to create a virtual 3D music world in which interaction with music in different ways is possible. MeXX is scheduled as a virtual environment based on 3D visualization techniques. The two main reasons why it was decided to use a 3D representation were on the one hand the general fascination which modern 3D environments exert and their certain attractiveness in participating and at experiencing them. On the other hand novel technologies that make it possible to establish 3D sound are also considered as being an active part within MeXX to provide a more intensive music experience.

In Chapter 2 existing systems were evaluated with respect to their purposes and technologies to get an overview of the state of the art. Therefore the research areas on Music Information Retrieval, Virtual Environments, Music-Based Games and Music Environments were consulted. Relevant systems which deal with music in various ways were selected and examined in detail.

3.1.1 Database Preparation

In parallel to the literature research own ideas for a 3D music world have been created. In a first step available music information probably consumed by users was summarized and different scenarios were drawn how music information can be experienced. The scenarios are shown in Table-2 and provided the base for that concept.

Sheet Music	…	tabs, chords, scores
Textual Information	…	title, lyrics, biographies
Images	…	CD covers, pictures of artists
Audio Files	…	live recordings, MIDI files
Descriptive Data	…	publication year, genre, mood, tempo, rhythm
Videos	…	music clips, concert videos
3D Models	…	of artists, songs or instruments, or of virtual stages

Table-2: Scenarios how music information can be experienced

To establish a repository of music information for a 3D world a database was created. The *MySQL Community Server* which is open source and can be downloaded for free from the MySQL homepage[42] was used. The database has the five tables,

- *MusicPiece*
- *Artist*
- *Album*
- *Genre*
- *Mood*

Figure-28 sketches the structure of the database. The table music piece is the central element and specifies the songs by the attributes `ID`, `title`, `year` of publication, `tempo` and `audiofile` which is the name of the file containing the audio data. An artist is described by his `name`, `year` (either the founding year of a band or the birth year for a single performer) and the file name of a `picture` of the performer or the band. The album table contains `ID`, `name`, `year` of publication and the file name for an album `cover`. The *genre* and *mood* tables contain only `name`.

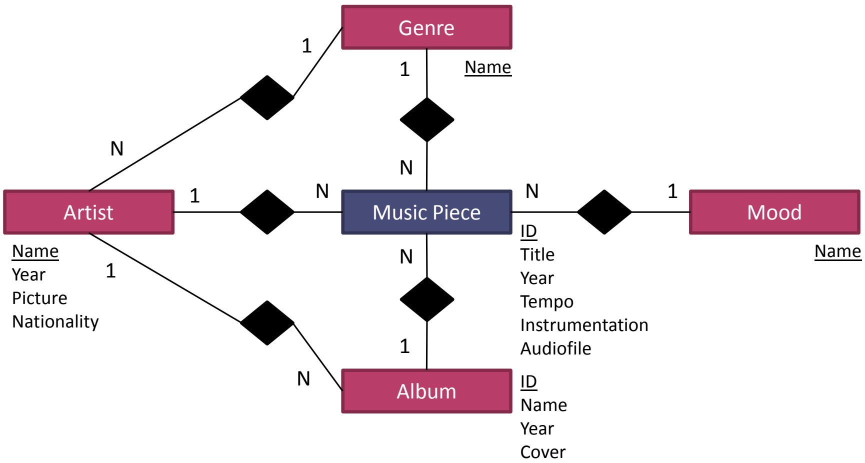


Figure-28: Database diagram

In a first approach the database was filled with 40 music pieces from 20 albums of 20 artists. The `genre` and the `year` were assigned to the music pieces according to the metadata in the audio file. The `tempo` for each music piece was manually determined and the `mood` was assigned subjectively. The *genre* table currently includes the features `Blues`, `Classical`, `Hip Hop`, `Jazz`, `Metal`, `Rock` and `Pop`; the *mood* table holds `Fear`, `Hostility Guilt`, `Sadness`, `Joviality`, `Self-assurance`, `Attentiveness`, `Shyness`, `Fatigue`, `Serenity` and `Surprise` as emotional descriptors. The audio files and images for CD covers and artist pictures are stored outside the database.

[42] MySQL Homepage: http://www.mysql.com/

3.1.2 Finalization of Concept

The design of music environment systems in terms of their contents and functionalities depend on the end-users interests and desires. Therefore three overlapping areas for music environments as sketched in Figure-29 were worked out.

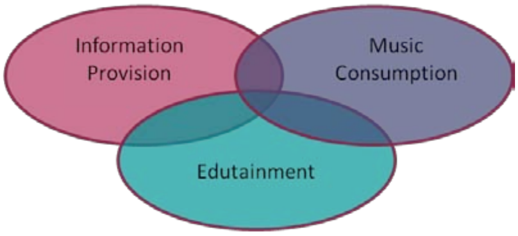


Figure-29: Possible areas for music environments

Environments for information provision focus to provide the user with specific information about music. Such information may be about a song like title, lyrics, audio file, genre, rhythm, sheet music, or about an artist including pictures or biographies, or about an album like track list or publication year. Possible scenarios are music repository browsers, virtual music stores and museums.

Systems for music consumption aim at the entertaining factor of music within a 3D world like sound collages, virtual concert simulations or virtual bars and dance clubs.

> *Edutainment is the act of learning through a medium that both educates and entertains.*
> *(The American Heritage Dictionary, 2004)*

Edutainment scenarios within music systems force the active participation of a user to learn about music. Examples are 3D sound collages to try out different sounds of instruments, 3D worlds for a collaborative music composition or music-based games.

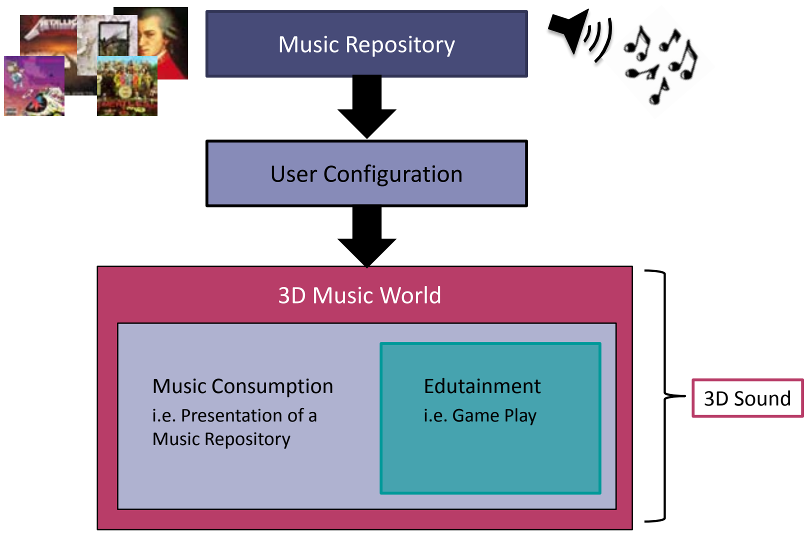


Figure-30: Concept of MeXX

As the last step in the concept development the areas music consumption and edutainment were considered to be included in MeXX, the systems implemented by Torrens et al. (2004), van Gulik et al. (2004) and Knees et al., (2007) served as a source for inspiration. This led to the decision to develop MeXX as a user-controllable 3D world that visualizes music repositories, plays 3D sounds and fulfills two purposes regarding to music consumption and edutainment:

- 3D visualization of a music repository for music exploration and consumption
- Game-setup for music examination

The resulting concept for MeXX is sketched in Figure-30 and described in more detail below.

3D Visualization of Music Repositories

The first purpose of MeXX is to create a 3D visualization of a particular music repository that follows the database design as presented in section 3.1.1. The repository contains information about music pieces, artists and albums presented in the 3D world by geometrical objects. The final decision was that the music pieces are the central element in the 3D world including the playback of their corresponding audio files as 3D sound.

The discrete and numerical features listed in Table-3 were chosen as candidates to define the position of a music piece and artist or album respectively within the 3D world. A pair wise combination out of those features can be used to build up the 3D world. They are mapped on the x- and z-axis that span the ground of the 3D world. The music objects are mapped into the 3D world according to the feature values.

Discrete Features		**Numerical Features**	
Genre	Mood	Tempo	Year

Table-3: Features

Customization of the music pieces to be displayed in the 3D world is enabled by a filter, for instance music pieces which only belong to the *genres* `blues` and `rock` and have a *tempo* between `80` and `100` bpm can be selected. Besides the arrangement and amount of music pieces, optionally a third feature can be used to color encode the music pieces in the 3D world - all music pieces that belong to a particular feature value are colored equally.

In the 3D world navigation is possible and the perception of 3D sound is established. Depending on the information a music repository contains, a variety of perceptions and insights into music may be entertainingly obtained. The following scenarios can be established by MeXX:

- The music repository contains a mix of music pieces of several artists. MeXX provides the user with a 3D visualization of that repository to perceive music sounds of several artists coming from several, different directions. If songs are similar in terms of the selected features they are located close to each other and similarities between songs can be recognized visually and aurally, at the same time.

- The music repository contains music pieces of a single artist. In this case the 3D world provided by MeXX acts as a “homage” to that artist since the user is purely presented with the artwork of that particular artist. In that case changes in an artist’s music style as well as “Ups and Downs” in his career are delivered in an entertaining way.

- The music repository contains different interpretations of a single music piece performed by various artists. Thus MeXX outlines different moods as well as tempo changes and genre switching the interpretations invoke.

- If a music repository only contains music pieces that are performed by specific instruments like piano, strings, or guitar MeXX can be used to monitor changes in the performance style of that particular instrument over the years or over different genres.

In addition MeXX is conceptualized in a way that all above mentioned scenarios can be extended to a game environment to force an intensive music examination.

Game Setup for Music Examination

The integration of a game logic is the second main part of the concept of MeXX. Playing a game, the goal is to score points by placing a number of music pieces to their proper positions within the 3D world. The concept of the game is described below:

- In the first step two features responsible for the layout of the 3D world are chosen. This selection implies that the user will be challenged on those two features.

- After the 3D world is created, the user gets a particular (random) number of songs out of the music repository. These songs are further called the outer songs, the features of them are hidden and only listening is possible.

- During the game phase, navigating through the 3D world allows to visually and aurally perceive the remaining music pieces out of the repository - this can be seen as hinting process since all remaining songs are positioned correctly in the 3D world. The user is forced to attentively listen to the outer songs to decide on their correct positions. In case of upcoming doubts replacement to another position is possible.

- At the end of the game, when all outer songs are positioned in the 3D world, the solution and the reached score are presented.

Within the game environment of MeXX the decision process on where to place a song forces the user to intensively listen to a music piece for feature identification. Games are (usually) played by people to entertain them and force their development of strategies. The aim of MeXX is to train the user's sense of music but hide the learning process which is a natural benefit.

3.1.3 Examination of Tools

To implement MeXX, technologies that serve as an appropriate implementation base for 3D music worlds were examined. Exploring *Sun's Project Wonderland* (Sun Microsystems, Inc., 2010) seemed to be an interesting solution since it is released with several functionalities to create a 3D world including the client-server architecture for multi-user purposes. But at that time (summer 2009) SUN announced a new release of Project Wonderland with re-implemented and improved technologies to be released a few months later. For this reason Project Wonderland in it's at that time recent version 4.0 at that time was put back for further examination.

Alternatively *Open Croquet* (Croquet Consortium, 2010) was taken into account. Experiments with the system in terms of content creation and usability were made. Its lack of documentation for supporting 3D sound, rather the *Torque Game Engine* (GarageGames, 2010) was considered for being the environmental base as it supports 3D sound and comes with a full implemented game-style environment.

More investigation on 3D tools led to *VRML/X3D* (Web3D Consortium, 2010) which is an interesting solution because of its standardization over the World Wide Web, its support of 3D sound and its comprehensive 3D modeling process following the scene graph model. In addition VRML/X3D worlds can be created or modified in runtime through common programming languages like Java or C/C++ which is an interesting feature. This led to the decision to completely switch to Java and its extension library Java3D (Sun Microsystems, Inc., 2010) since it additionally supports 3D sound, the loading of external models and can be integrated or extended by any customized Java application.

3.2 Java3D Implementation

For the implementation of MeXX Java3D was chosen in its recent release version 1.5.2. Java3D is a standard extension to the Java2 JDK and provides a collection of high-level constructs for the creation, rendering and manipulation of 3D scenes. Java3D programs can be either stand-alone applications or web-based applets.

Java3D uses the OpenGL rendering engine by default, but also the DirectX engine can be activated by a command-line argument. DirectX and OpenGL are almost equivalent since both are based on the same graphics pipeline. The most significant difference between the two APIs is that OpenGL supports a wide range of platforms and operating systems whereas DirectX is limited to Windows and the XBox. JOGL is a thin layer of Java over OpenGL and is used to combine OpenGL and Java3D more closely. Java3D separates the virtual and physical worlds to make it straightforward to reconfigure an application supporting different output devices like monitors, eyeglasses with stereo displays or CAVE's.

The Java3D API defines over 100 classes available in the `javax.media.j3d` package. These classes are commonly referred to as the Java 3D core classes. In addition to the core classes Java3D uses two more packages. The `com.sun.j3d.utils` package that is referred to as the Java3D utility classes provides additional functionalities to the core classes like content loaders, scene graph construction aids and geometry classes. The `javax.vecmath` package defines vector math classes for points, vectors, matrices, and other algebraic objects.

The Java3D SDK can be downloaded from the project homepage[43] and includes all necessary classes for a successful Java3D development. The website provides also several tutorials and sample programs, but in addition the references of Davison (2005 and 2007) were used; they are a comprehensive reference in terms of game programming in Java3D.

3.2.1 The Java3D Scene Graph

Java3D defines the concept of a virtual universe as a 3D space with an associated set of objects. These objects are arranged inside the virtual universe following the scene graph model. Advantages in using a scene graph model are that it hides low-level 3D graphic elements and thus simplifies 3D programming and speeds up the resulting codes. But optimization tasks like view culling, occlusion culling, level of detail selection, and behavior pruning must be coded in lower-level APIs.

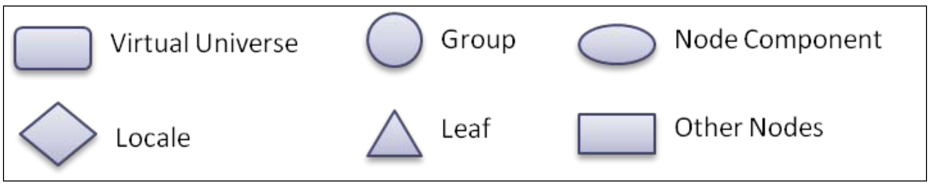


Figure-31: Standard symbols representing objects in a Java3D scene graph

To design a Java 3D virtual universe a scene graph is drawn using a standard set of symbols. That graphical representation of a scene graph can serve as design tool and/or documentation for Java3D programs. After the design is complete, that scene graph is the specification for the program and a

[43] Homepage of Java3D Project: https://java3d.dev.java.net/

concise representation of it. Figure-31 shows the typed symbols representing objects in the scene graph.

VirtualUniverse and *Locale* are specific nodes that build up the root of a 3D scene and are always necessary when creating such a 3D scene; they are described in more detail later. A *Group* node has child nodes and groups them in a way that operations like translation, rotation and scaling can be applied en masse. It supports the positioning and orientation of its children and is sub-classed to extend the operations *BranchGroup* and *TransformGroup*. The *BranchGroup* operation allows children to be added or removed from the scene graph at runtime and the *TransformGroup* operation changes the position and orientation of its children.

Leaf nodes represent visible items within the virtual universe such as 3D objects but also non-tangible things like lights, fog, behaviors or sounds. *Node Components* define the attributes of the *Leaf* nodes and are referenced by them. They influence the appearance like color, material, reflectivity and the geometry of a 3D object. *Other Nodes* represent objects which are necessary for the definition and determination of the position, orientation and size of a viewer within the virtual world. They are also necessary for the output of the 3D world on the screen.

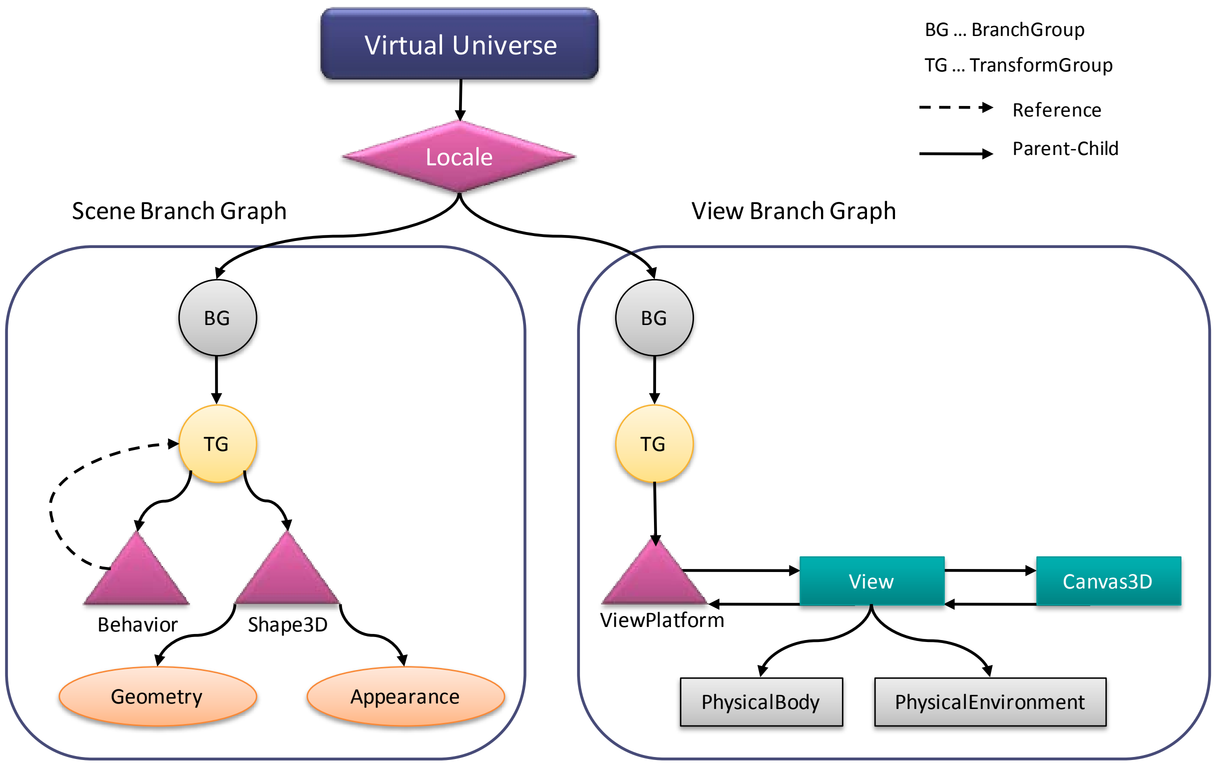


Figure-32: A typical scene graph structure within a Java3D application

Figure-32 sketches the scene graph structure of a typical Java3D application represented by the standard symbols. The scene graph consists of four main parts, the *VirtualUniverse*, the *Locale*, the *Scene Branch Graph* and the *View Branch Graph*:

- The *VirtualUniverse* is the top-level class of all objects. It is the root of the scene graph and represents the virtual world space and its coordinate system. It defines a named universe and consists of one or more *Locale*.

- The *Locale* acts as the scene graph's location in the virtual world. It has no parent in the scene graph but is implicitly attached to the *VirtualUniverse* when it is constructed. It contains a list of scene graph nodes (*BranchGraphs*).

- The *Scene Branch Graph* in Figure-32 is the left branch of the *Locale*. It holds the virtual world's content like 3D objects with their geometry, material, texture, appearance, light, fog, sounds and the world's background. Furthermore the *Behavior* of 3D objects like animations are located within the *Scene Branch Graph*.

- The *View Branch Graph,* the right branch below *Locale*, specifies the users positions, orientations and perspectives as they look into the virtual world from the physical world; for instance in front of a monitor. The *ViewPlatform* in the *View Branch Graph* controls position and orientation of the viewer in the virtual world, the *ViewPlatform* locates a *View* within a scene graph. The *View* is the main view object that contains all parameters required in rendering a 3D scene from a viewpoint, the location and orientation of a user within the virtual world. The *Canvas3D* is the 3D version of the AWT Canvas object representing a window in which Java3D draws images. The *PhysicalBody* is an object that contains calibration information about the viewer's physical body, and the *PhysicalEnvironment* contains the physical specification about the environment in which the view will be generated. It is used to set up input devices like sensors for head-tracking and other uses, and the audio output device.

3.2.2 Setup of the 3D World

In a first step the basic environment for a 3D world in MeXX was created. Figure-33 shows the standard recipe recommended by SUN to create all basic and necessary components of a virtual Java3D world. First a *Canvas3D* object needs to be created to provide a drawing base for the world; it was integrated within standard Swing GUI components including *JFrame* and *JPanel*.

The top root *VirtualUniverse* and *Locale* are instantiated and connected in the next two steps. In the fourth step the *View Branch Graph* is constructed with its necessary components to manage frame drawing of the 3D world. Next the *Scene Branch Graph* is created to fill the world with 3D models, lights, etc.; this step is illustrated in more detail below. Afterwards *View Branch Graph* and *Scene*

Branch Graph are compiled with a command due to performance reasons and the two branch graphs are inserted into the *Locale* to make them "alive".

After these steps, the *Scene Branch Graph* was extended with a blue *Background* and a green floor (constructed by 4-point *QuadArray*). For enlightening the 3D world three light sources were implemented; one *AmbienLight* that comes from all directions but does not force diffuse or specular reflections on object surfaces, and two *DirectionalLights*. The *DirectionalLights* are bundled lights with their origin in infinity contributing diffuse and specular reflections.

1. Create a *Canvas3D* Object.
2. Create a *VirtualUniverse* Object.
3. Create a *Locale* Object and attach it to the *VirtualUniverse* Object.
4. Construct a *View Branch Graph*:
   a. Create a *View* Object.
   b. Create a *ViewPlatform* Object.
   c. Create a *PhysicalBody* Object.
   d. Create a *PhysicalEnvironment* Object.
   e. Attach *ViewPlatform*, *PhysicalBody*, *PhysicalEnvironment*, and *Canvas3D* Objects to *View* Object.
5. Construct *Scene Branch Graph(s)*. ← 3D world is designed here.
6. Compile *Branch Graph(s)*.
7. Insert Subgraphs into the *Locale*.

Figure-33: Recipe for writing Java3D programs

The resulting 3D world is shown in Figure-34. It is worth to mention that after these steps there was still no navigation implemented; this was done separately by choosing an appropriate navigation style.

Figure-34: Basic environment of the 3D world

Establishing the Navigation

For navigation in a 3D world, Java3D provides the class *KeyNavigatorBehavior* that allows modifying the user position within the 3D world during runtime. Movement though the world is initialized by pressing particular arrow buttons on the keyboard.

The *Java3D Code Repository* website[44] provides a collection of useful utilities for Java3D programming. This includes also a navigation utility package providing handlers for both, mouse and keyboard input as well as more dynamic capabilities such as terrain following and collision detection. A mouse navigation is established by the *MouseViewHandler* by click and drag actions; a move is initialized by pressing the mouse button and dragging the mouse in a desired direction. A small drag causes slow movement while a large drag causes fast movement. The *MouseViewHandler* supports several navigation modes:

- *None*: requests from mouse or keyboard are ignored and thus all navigation is disabled.
- *Examine:* the viewpoint is rotated around the local origin while examining the 3D world.
- *Fly*: movement through the world in forward, reverse and lateral direction is enabled. Collision detection is available but no terrain following.
- *Walk*: movement through the world in forward, reverse and lateral direction is enabled while bound to the terrain. Collision detection is available.
- *Pan*: the camera moves along the local plain axes. There is no collision detection or terrain following available.
- *Tilt*: the camera is anchored and rotates in an up/down and left/right fashion. There is no terrain following or collision detection available.

In MeXX the *MouseViewHandler* of the Java3D Code Repository was implemented since the navigation movements are smoother than those of the standard *KeyNavigatorBehavior* and it allows speed control of movements. To establish the navigation the *MouseViewHandler* needs to be attached to the *View Branch Graph*. The left mouse button is responsible for *walk*, the middle button for *pan* and the right button for *tilt*. Additionally switching between all navigation modes during runtime is supported by a drop-down menu.

Content Design

The next step was to fill the world with 3D objects. Therefore appropriate 3D representations for the music pieces, artists and albums were designed. Java3D supports the import of external 3D models and comes inherently with two loader classes to integrate external 3D scenes in the *Lightwave* (`.lws`)

[44] Java3D Code Repository: http://code.j3d.org/

or *Wavefront* (.obj) format. In addition SUN provides the VRML97 loader[45] for integration of VRML97/2.0 (.wrl) scenes into a Java3D world. CyberVRML79[46] is another loader supporting the import of .wrl files as well as X3D (.x3d) models, the AC3DLoader[47] loads model information in the AC3D (.ac) format.

Experiments with above mentioned loaders were made to integrate external 3D objects such as .wrl or .obj models created in common 3D modeling programs like Blender or Maya. But finally the use of external 3D models was disclaimed since Java3D already provides numerous appropriate 3D primitives suitable for MeXX. The *Sphere*, *Box*, *Cone*, *Cylinder*, *Text2D* and *Text3D* objects are ready for use by setting attributes like color, texture, size or position after instantiation. In MeXX several types of such 3D primitives have been customized for the creation of the 3D music objects shown in Figure-35.

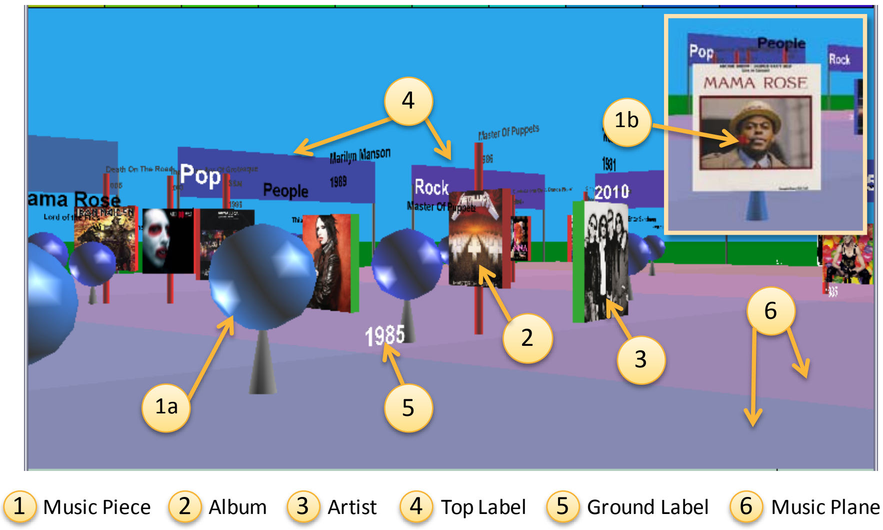


Figure-35: The 3D music objects which appear in the 3D world

(1) A *MusicPiece* can be represented by the following layouts: **(a)** the sphere layout consists of a colored *Cone* and a *Sphere* on top of it, **(b)** the album layout consists of a colored *Cone* and *Box* and maps a *Texture* which uses a corresponding bit map file (.jpg) of the album on two sides of the *Box*. In both cases a *Text2D* object is placed on top of the *MusicPiece* that specifies its title. Additionally it consists of an invisible *PointSound* object that generates the sound.

[45] VRML79Loader form SUN: https://j3d-vrml97.dev.java.net/
[46] CyberVRML79 Loader from CyberGarage: http://www.cybergarage.org/
[47] AC3DLoader from j3d.org Code Repository: http://code.j3d.org/impl/ac3d_loader.html

(2) The *Album* object consists of a red *Cylinder*, a red *Box* and a *Text2D* object indicating the album's `artist` and the `year` of publication. In addition the album cover is mapped to the *Album* object as a *Texture*. The *Album* objects only appear if *genre* and *year* were chosen as axes features.

(3) The *Artist* object is designed as a *Box* with a green frame and a *Texture* showing the picture of the artist. A *Text2D* on top of it indicates the artists `name` and the founding `year` of the band or the birth `year` of a single performer. Like the album objects, they are only displayed in the 3D world if *genre* and *year* were chosen.

(4) The *TopLabels* are built by two *Cylinders*, a colored *Box* and a *Text2D* instance and show the feature values along the feature axes and thus support orientation within the 3D world.

(5) The *GroundLabels* have a similar functionality as the *TopLabels* and show the feature values of both axes additionally in the music grid. They are simply *Text2D* objects.

(6) The *MusicPlanes* are floor elements and build the *MusicGrid* (cp. Figure-37). They are constructed by colored *QuadArray* instances and thus support orientation within the 3D world in a further way.

Java3D provides *Behavior* classes; they are added to the scene branch graph to define specific user actions like navigation. *Behavior* classes support for example translation, rotation and zooming of scene graph objects. Within MeXX the *PickRotateBehavior* and *PickTranslateBehavior* are added to the *Scene Branch Graph* to enforce the user's interactivity and control within the 3D world. The *MusicPiece*, *Album* and *Artist* objects are all stored within their own *TransformGroup*. Setting appropriate permission capabilities enables a runtime transformation of them; these objects can be moved or rotated by mouse operations.

If the album layout for music pieces is used, a *BillboardBehavior* is attached to the object. This behavior is used to automatically cause the front textured plane to always point at the viewer's eye position when getting close enough.

Dynamic Content Creation

Since the concept of MeXX is scheduled to support customization of the 3D world layout and its objects, a start GUI was implemented as an input platform. The start GUI shown in Figure-36 enables the choice of several options, whereas

- Feature matrix,
- Axes scaling,
- Color selection and
- Appearance selection

are the main controls to define the visual layout of the 3D world. A music piece has four features which are relevant for the appearance of the 3D world,

- *Genre*
- *Tempo*
- *Mood*
- *Year*

The feature matrix enables the selection of two of those features. These two features – denoted furthermore as axes-features – are mapped to the axes that span the ground of the 3D world and thus serve as an arrangement base for the music pieces. The axes are partitioned into sections and the scaling factors define the size of them in virtual meters. The factors are used to transform the feature values of the music pieces to distance values in the 3D world. The color selection enables another feature to serve as a base for the color encoding of the music pieces. The appearance selection defines the geometrical layout of the music pieces; either the colored sphere layout or the textured album layout. In Figure-36 *genre* and *year* are selected as axes features and *year* is selected as base for the color encoding. The sphere layout was chosen to represent a music piece and for the axes sections a scaling factor of 10 was used.

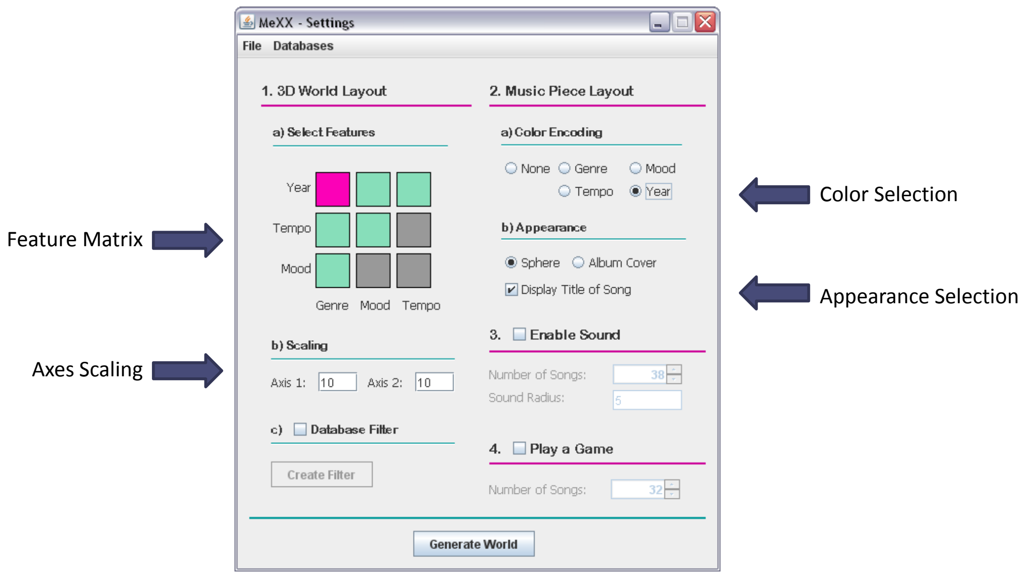


Figure-36: Start GUI with its feature matrix and options

After the selection is established the 3D world is built up by pushing the "Generate World" button. At the first step so called visual containers are created, their purpose is to provide the user with context information and to support orientation within the 3D world. The visual containers are the *TopLabels*, the *MusicPlanes* and the *GroundLabels*.

The *TopLabels* are positioned along the axes which are partitioned into sections according to their mapped features. In case of discrete feature values like *genre* or *mood*, the sections are represented by the feature values itself. In case of numerical values as *tempo* and *year*, appropriate section sizes are computed. Therefore for each axes the numerical range of the features in the database is computed as the absolute difference of the ceiling of the maximum and the floor of the minimum value. The section size itself is generated by partitioning the data range evenly as shown in Figure-37.

The sections belong to the particular axes and make up the so called music grid. They are visualized in the 3D world by the *TopLabels* whereby they are additionally color encoded by a section color. The first section of an axis has a particular starting color and the color of subsequent sections is slightly changed by increasing the hue depending on the number of sections. Therefore every *TopLabel* has a different color.

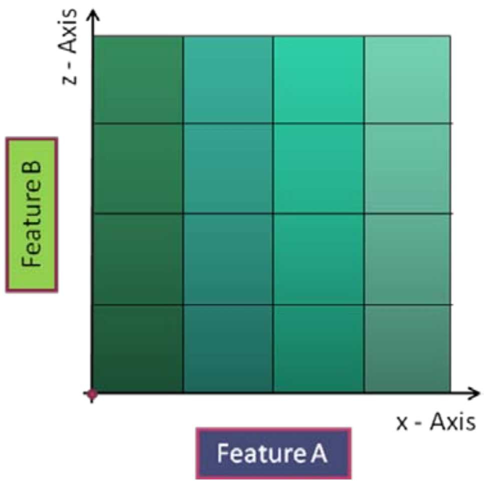


Figure-37: The music grid

In the next step the *MusicPlanes* are created to border the x-, and z-sections within the music grid and thus to facilitate the user's orientation. One *MusicPlane* fills the area between one x- and one z-section with a solid color. Again a starting color is initialized that specifies the hue, saturation and brightness of the first *MusicPlane* which is normally generated from the first x- and z-section. For each claimed x-section the color of the corresponding *MusicPlane* changes by subsequently increasing the hue of the starting color. Along the z-axis the color is changed by subsequently increasing the brightness. Thus the change of hue and brightness on the floor supports a subjective perception of the position when navigating through the 3D world.

After the visual containers are created, the *MusicPieces* are inserted into the world. The position of a *MusicPiece* is thereby depending on the selected axes features and the size of the objects related to the world by the axes' scaling factors; the larger the scaling factor the smaller the objects appear in the world. In case of numerical feature values, the position of a *MusicPiece* is determined by applying a linear transform to them. In case of discrete feature values the *MusicPieces* are positioned randomly in the corresponding section. The color of a *MusicPiece*, which is to be more precise the color of the sphere if the sphere layout was chosen or the color of all un-textured elements of the album layout depends on the selected color feature. The *MusicPieces* are color encoded by colors estimated in the same way as they were already computed for the *TopLabels*.

Figure-38 shows the resulting 3D world using *genre* and *year* as axes-features and *year* as color-feature. The *genre* is mapped to the x-axis (E-W) and the *year* to the z-axis (N-S). For color encoding *year* is chosen as color-feature and all music pieces that belong to the same z-section as well as the corresponding *TopLabel* have the same color.

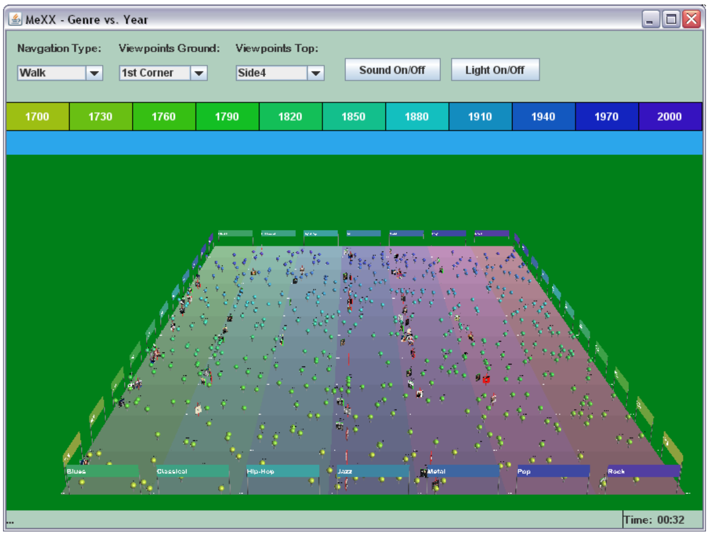


Figure-38: Visual layout of the music grid in the 3D world

Furthermore a night simulation within the 3D world was established to force the fun factor in exploring the 3D world. Therefore the background color can be changed to black during runtime and additionally a black *LinearFog* and a *PointArray* containing a star-map is attached to the scene branch graph. Figure-39 sketches the resulting appearance of the 3D world.

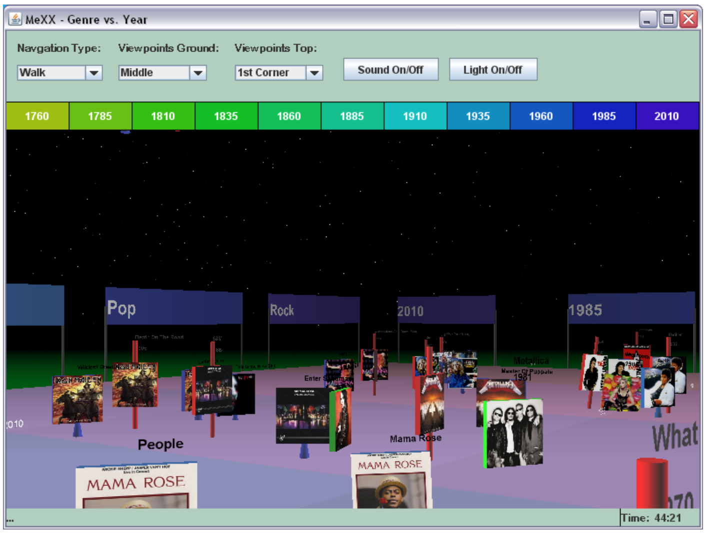


Figure-39: Night simulation using a *LinearFog* and a *PointArray* as a star-map

Game Setup

Within the start GUI of MeXX, the user can select the option to play a game. In that case the user is presented with a sub-set of music pieces out of the music repository. The goal of the game is to position these music pieces correctly in the 3D world. To setup the game environment again that two axes features need to be selected as well as a color encoding and a preferred appearance layout for the music pieces. Additionally the user specifies the number of outer songs, to be challenged on. The 3D world is build up as already shown above; only the outer songs are excluded, they need to be inserted into the world by the user. The outer songs are randomly selected out of the underlying music repository and presented to the user as red spheres within a separate panel on the right side of the 3D world GUI. The resulting game environment is sketched in Figure-40, 15 outer songs have been chosen.

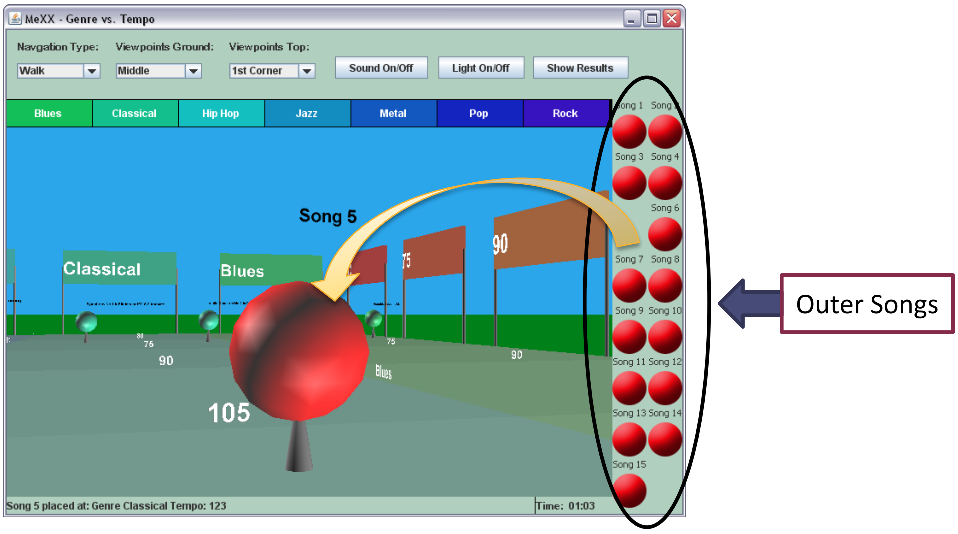


Figure-40: Game environment of MeXX

MeXX stores the feature values of the outer songs internally for comparison with the user's choice. To play the outer songs Java's *AudioClip* implementation has been used whereas the left mouse button is set to be responsible for playing and stopping the sound.

A Java procedure monitors all operations during the game. The right mouse button controls the insertion of an outer song; a click on the red sphere at the user's current position forces the calculation of the coordinates. That coordinates are used to place the outer song and to retrieve the feature values for that position. After all outer songs are inserted the resulting feature values are compared to the true feature values to calculate the score.

In case of discrete feature values the score is increased by 1 point only if estimated and true feature values are identical. In case of numerical feature values two tolerance boundaries are used. The size depends on the numerical range of that feature in the database or the resulting numerical range if a filter was applied. The large boundary is a tenth of that numerical range and the small boundary is the half size of the large one. If an outer song was assigned within the smaller boundary, the score is increased by 1 point. Placement within the larger boundary still raises the score for a ½ point. For example in Figure-40 the music pieces are arranged in terms of *genre* and *tempo* whereas the *tempi* range from 60 to 210 bpm; the small boundary for *tempo* is thus 7.5 and the large one is 15. The fifth outer song was placed at *genre* classical and *tempo* 123 bpm. The true feature values for that song are the *genre* classical and *tempo* 135 bpm. For the correct assigned *genre* the user obtains 1 point. The distance from the assigned *tempo* (123 bpm) to the correct *tempo* (135 bpm) is 12 bpm, thus the user scores ½ point since the *tempo* was positioned outside the smaller and inside the larger boundary. Therefore the user scores 1½ points for that song in total.

At the end of the game the results appear in a separate frame and are stored in a text file to make a later examination possible - more details are presented in Chapter 5. The upcoming Chapter 3.2.3 explains how 3D sound is established in MeXX.

3.2.3 Management of 3D Sound

Java3D supports spatial sound and background sound playback. A background sound is a non-spatial sound source without position and direction. The sound is played without modifications and useful for playing a mono or stereo music track. It is used to generate ambient sound effects with more than one background sound concurrently enabled and activated. A 3D sound source is spatially located within the 3D world. Depending on their relative positions users perceive the sound as being left or right, behind or in front, above or below it.

Java3D provides two classes for generating 3D sound effects: *ConeSound* and *PointSound*. The *ConeSound* emits sound in one particular direction whereas the *PointSound* radiates the waves uniformly in all directions. In MeXX the *PointSound* is used since it produces a more realistic music playback effect similar to real world.

Audio Mixer

To play a 3D sound correctly an appropriate audio device is required to manage the sound volume and sound position by audio mix. Since the *JavaSoundMixer,* which inherently comes with Java3D, does not support spatial audio playback, the JOALMixer published in 2006 by the Java3D community member David Grace was used.

The JOALMixer is a JOAL[48]-based audio device and can be downloaded from the Java3D downloads page[49]. JOAL is a Java-binding for OpenAL and implemented within Sun's JOAL Project. The aim is

[48] JOAL – **J**ava **O**pen **A**udio **L**ibrary

to provide hardware supported 3D spatial audio for games written in Java. OpenAL is a cross-platform 3D audio API that models a collection of audio sources located in a 3D space and heard by a single listener somewhere in that space. According to its specification[50] OpenAL only spatializes sounds with one channel that come in the `.wav` format; audio buffers containing audio data with more than one channel will be played without 3D spatialization features since these formats are normally used for background music. Figure-41 sketches the basic steps for a sound being rendered as a 3D sound within a Java3D application.

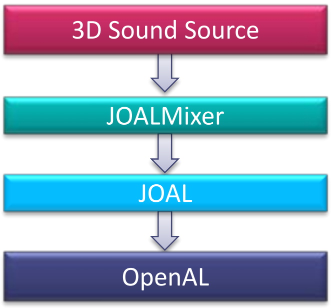


Figure-41: Audio pipeline to produce 3D sound

To run the JOALMixer, OpenAL drivers must be installed; they can be downloaded for Windows, Linux and MacOS from the *Creative Labs* homepage[51]. Additionally JOAL must be installed, it can be downloaded from the JOAL Project homepage[52]; the file contains two JAR files, `joal.jar` and `gluegen-rt.jar` and two DLL's, `joal_native.dll` and `gluegen-rt.dll`. The JARs need to be included in the program's class path, the two DLLs can be placed in any directory but at runtime the path to that directory needs to be supplied.

```
// --- Setup JoalMixer as an AudioDevice
JoalMixer audioMixer = new JOALMixer(physicalEnvironment);
audioMixer.initialize();
physicalEnvironment.setAudioDevice(audioMixer);
```

Figure-42: Initialization of JOALMixer

To configure the JOALMixer as audio device for the 3D world, a new instance needs to be created and associated to the physical environment accessing the audio device; Figure-42 shows the Java code for these operations. Once the JOALMixer is initialized as audio device, the application can additionally select the audio playback type as output for the rendered sound - valid playback devices are head-phones or speakers.

49 Java3D Downloads Page: https://java3d.dev.java.net/binary-builds.html
50 http://connect.creativelabs.com/openal/Documentation/OpenAL%201.1%20Specification.htm
51 http://connect.creativelabs.com/openal/default.aspx
52 SUN´s JOAL Project: https://joal.dev.java.net/

Sound Behavior

Each *MusicPiece* in the 3D world has a *PointSound* attached for playing the corresponding audio file. When a *PointSound* is created and enabled and has a reference to an audio file the JOALMixer directly loads the audio file in its internal buffer for mixing purposes. It is worth to mention that the JOALMixer in its recent version 1.5.2 is not yet completely implemented as outlined by the author but nevertheless works well in producing correct 3D sound as the following two experiments have shown:

(1) All *PointSounds* were created and enabled and had a reference to an audio file already before the program was executed.

(2) All *PointSounds* were created but **not** enabled and **not** referenced to an audio file until the sound was required at runtime (i.e. when the users approaches a *MusicPiece*).

In the first approach the *PointSounds* are created from scratch and all audio files are buffered into memory when the program is started. Thus an observable waiting period occurs until the 3D world is ready to use. Unfortunately the JOALMixer stops the buffering after loading more than 16 sounds due to not clearly identifiable reasons with the result that all *PointSounds* created later did **not** play any more sound.

In the second approach the *PointSounds* are again created from scratch, but enabling and referencing of the audio files occurs not until the user approaches close enough to a particular *MusicPiece*. The advantage using this approach is that there is no waiting period and that more than 16 sounds are played by the JOALMixer. But since the audio files are loaded in runtime a short bucking in frame rendering occurs when the user position forces loading an audio file.

The second approach was chosen as an appropriate solution for MeXX and was refined in some entities. Figure-43 sketches the strategy for combining *PointSound* and JOALMixer to play a sound.

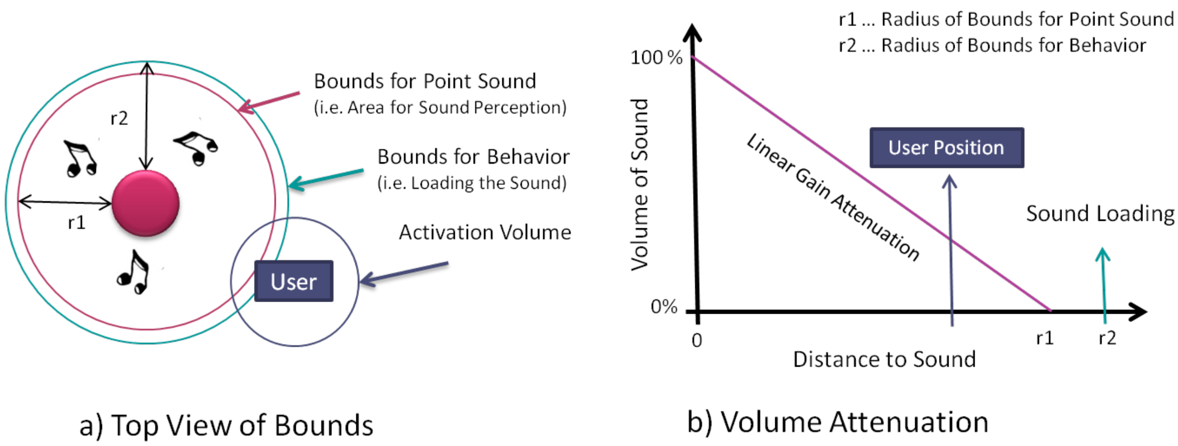


Figure-43: 3D sound management using *PointSound* and *Behavior*

(1) The center of the *ViewPlatform* which is equivalent to the user's position within the 3D world is surrounded by an *ActivationVolume*. This *ActivationVolume* which is actually an activation radius intersects with so called scheduling bounds of other objects when the *ViewPlatform* approaches to such objects.

(2) The *PointSound* has bounds to schedule the sound playback. For defining appropriate bounds a *BoundingSphere* with pre-defined radius is used. If the *ActivationVolume* intersects the bounds of the *PointSound* the sound playback is started; the closer the user to the *MusicPiece* the louder the sound is played. Linear gain attenuation as sketched in Figure-43b is used therefore since it is the only attenuation function by *PointSound*. Leaving the bounds of the *PointSound* the sound is stopped.

(3) To manage enabling/disabling of sounds an additional *Behavior* that also has a *BoundingSphere* is attached to the *MusicPiece*. A *Behavior* in Java3D allows user-defined actions to be executed when the *ActivationVolume* of the user intersects the bounds of that *Behavior*. Such a *Behavior* was utilized to control the sound loading; stepping into the behavior bounds enables the *PointSound* and the corresponding audio file is referenced for being loaded and played, the *PointSound* becomes active.

Since the JOALMixer loads all sounds directly into its internal buffer an additional mechanism had to be implemented to prevent a memory overload. First the audio files were shortened to about one minute clips and converted to mono `.wav` format resulting in a file size of about 5 MB for each song. A so called sound cleaner was implemented with the purpose to remove irrelevant audio data from memory. The cleaner is called if more than a predefined number of audio files have been loaded, it calculates the distances from the *ViewPlatform* to all active *PointSounds,* the most far away *PointSounds* are disabled and the sound references are deleted.

3.3 Summary

This chapter showed the development of MeXX. At the beginning section 3.1 sketched the steps in establishing the concept. Starting with the literature research to examine state of the art systems that deal with music, scenarios were worked out how music information can be consumed. A database was designed that holds information about music pieces, albums and artists. In a further step several tools to implement the 3D music world were examined and Java3D was chosen. Finally the areas music consumption, edutainment and information provision were worked out as possible application scenarios for music environments. MeXX was conceptualized as a customizable 3D visualization tool for a music repository that meets music consumption and edutainment requirements.

MeXX can be optionally initialized as a game environment to train the user's appreciation and sense for musical concepts in an entertaining way. The features *genre*, *mood*, *tempo* and *year* were chosen as candidates to define the arrangement of the music pieces in the 3D world. In section 3.2 the implementation steps using Java3D were presented. First all necessary objects for a Java3D *VirtualUniverse* were created within standard GUI Swing components and the navigation setup was done. Later the 3D objects *MusicPiece*, *Album*, *Artist*, *TopLabel*, *GroundLabel* and *MusicPlane* were designed using Java3D standard primitives, including *Behaviors* to rotate and translate them in runtime. Depending on the selected axes- and color features, the scaling factors for the axes, the color feature and appearance layout for the *MusicPieces* the 3D world is built up. Starting MeXX as game a custom number of so called outer songs are randomly selected from the music repository. They are displayed as red spheres on the right side of the 3D world GUI; listening to them is possible. Playing a game, the goal is to score points by placing the outer songs to their proper positions within the 3D world. The program records the user actions while inserting the outer songs to finally calculate the score.

The sound management in MeXX uses JOALMixer as an audio device and the Java3D *PointSound* class to configure the spatialized sound. A 3D sound source, the JOALMixer, the JOAL layer and the OpenAL library define all necessary audio processing steps to produce a 3D sound. The audio files are loaded into the 3D world in runtime and removed in an appropriate way to prevent from memory overload. The next chapter presents MeXX in all its facets and features as well as several screenshots.

4 Results of MeXX

In this chapter MeXX is explained in terms of usage and application scenarios are presented. Chapter 4.1 explains the Graphical User Interface (GUI) of MeXX and its utilization. Since MeXX is conceptualized to visualize a music repository and to be set up as a game environment, scenarios out of both areas are presented in Chapter 4.2.

4.1 Graphical Composition und Utilization of MeXX

To run MeXX one or more music repository(s) that follow the database structure as shown in Chapter 3.1.1 are required. Therefore a text file is used to configure the database connection and the directories of the corresponding audio files, album covers and artist pictures. The file name is `DB.config`, the content structure is sketched in Figure-44. Numerous databases can be referenced in the `DB.config` since MeXX allows concurrent processing of different databases.

```
// --------------------------------------------------
// --- Database Connection(s)
// --------------------------------------------------
// 1. Any Description or Name of Database
// 2. Connection String of Database
// 3. Path to Audio Files
// 4. Path to Album Images
// 5. Path to Artist Images
// --------------------------------------------------
// --- Database 1
My Music DB
jdbc:mysql://localhost:3306/mymusicdb?user=root&password=
C:/DB_Sources/Audio_WAV/
C:/DB_Sources/Album_Covers/
C:/DB_Sources/Image_Artist/
// --------------------------------------------------
⋮
```

Figure-44: Database definition

After program start the `DB.config` file is processed by MeXX to determine the database attributes; connections to the databases are established and the attributes are retrieved and checked for further usage.

The GUI of MeXX consists of four frames:

- Start GUI: provides options to configure the appearance of the 3D world
- Filter GUI: optionally allows to apply a filter to the music repository
- 3D world GUI: consists of the actual visualization of the virtual world
- Game results GUI: appears when the game is finished and shows the results

After establishing the database connection, the start GUI as shown in Figure-45 appears. One out of the databases listed in the “Databases” menu can be selected. The start GUI has basically six options to configure the 3D world.

- The feature matrix allows the user to select a pair combination of the features *genre*, *mood*, *tempo* and *year* which are mapped on the axes of the 3D world.
- Through the axes scaling the size of the axes sections can be adjusted. The greater the scaling factors, the larger the sections.
- Optionally a filter can be applied to the database to customize the music pieces to be displayed in the 3D world by defining specific feature values. The filter GUI is presented below.
- The color option allows the selection of an additional color feature.
- Within the appearance selection the user can choose between two music piece layouts. The sphere layout which basically consists of a colored sphere and the album layout. In the album layout the corresponding album cover is mapped on two sides of a colored box. Both layouts include optional text information of the music pieces’ titles.
- The sound option enables the playback of the music pieces’ audio files as 3D sound. Additionally the number of concurrently played songs and their perception bounds can be adjusted.
- The game option allows the user to play a game by specifying a number of songs to be challenged on.

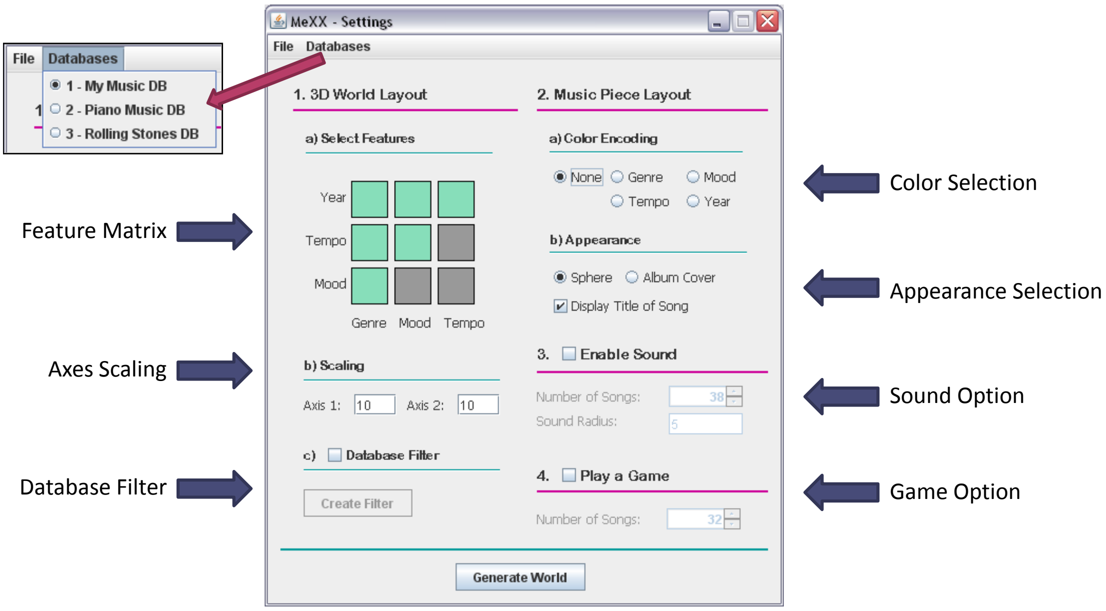
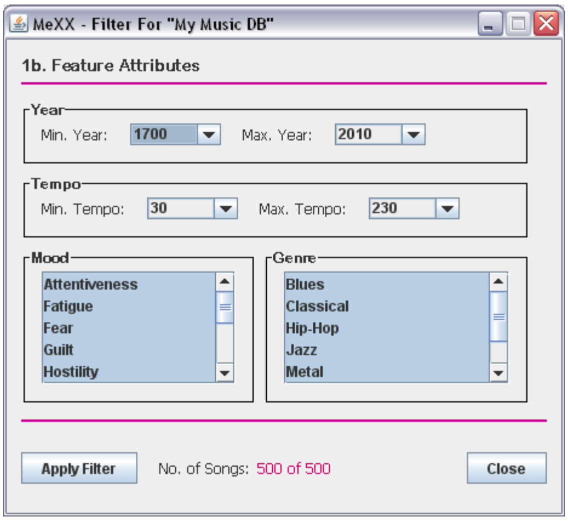


Figure-45: The start GUI

The filter GUI of MeXX is sketched in Figure-46 and consists of four frames according to the features *genre*, *mood*, *tempo* and *year*. Depending on the selected database (i.e. the active database), the entries in the drop-down menus and the lists are directly supplied from the database. Thus the filter GUI reflects the content of the actual database and makes an examination possible. Pressing the "Apply Filter" button immediately shows the resulting number of music pieces to be displayed in the 3D world.



<u>Figure-46:</u> The filter GUI

Pressing the "Generate World" button in the start GUI, the 3D world GUI appears on the screen. Depending whether or not the game option has been chosen

- the visualization layout or
- the game layout

is created.

Visualization Layout

The visualization layout of the 3D world GUI consists basically of four separate panels. They are depicted in Figure-47 and are the

- Control panel,
- Color panel,
- 3D world panel and
- Output panel.

The control panel provides options to manipulate the 3D world. Through the drop-down menu "Navigation Type" the user can select one of the modes *walk*, *pan*, *tilt* or *examine*. Initially the left mouse button is set to *walk* and the middle one is set to *pan*. The drop-down menus "Viewpoints Ground" and "Viewpoints Top" provide pre-calculated viewpoints within the 3D world that quickly will change the position of the user within the 3D world. 14 viewpoints were implemented that surround the music grid. They are sketched in Figure-48 whereas the ground viewpoints are located at the corners and in the middle of the music grid and numbered from (G0) to (G4), their corresponding top viewpoints are indicated through (T0) to (T4). There are four more top viewpoints located above the sides of the music grid and denoted by (S1) to (S4). The initial user position when entering the 3D world is (G1). The top viewpoints are useful to get a first impression and an overview of the established 3D world in terms of the size of the music grid and its contents. Through the "Sound On/Off" button the user can control the sound playback, but only if the sound was initially enabled in the start GUI. Additionally the "Light On/Off" button can be applied to change the light conditions within the 3D world. It establishes a night simulation to effect a new impression and increase the fun factor.

Within the color panel the scheme of the color encoding of the music pieces is drawn as an additional program feature.

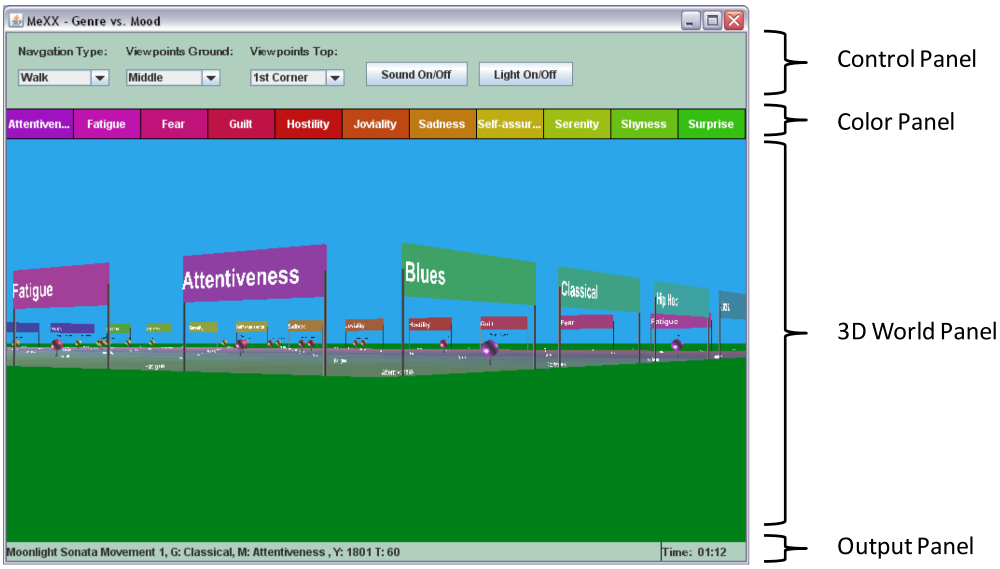


Figure-47 The 3D world GUI, visualization layout

The 3D world panel consists of the actual Java3D world that visualizes the music repository by use of the music grid. Mouse navigation is possible by pushing and dragging the left mouse button and 3D sound is produced. Additionally the user can interact with the music pieces: clicking on a music piece with the left button its features are sent to the output panel; additionally a rotation of that music piece is possible. Using the right mouse button a music piece can be repositioned.

The output panel reflects particular user actions. On the left side additional descriptions of a music piece are shown including *title*, *genre*, *mood*, *tempo* and *year* when the user clicks on it (the abbreviations "G" for *genre*, "M" for *mood*, "T" for *tempo* and "Y" for *year* are used in the panel). It is useful to retrieve that information of a music piece located far away from the user's current position but descriptions are rapidly desired. On the right side the elapsed time of the user within the music world is shown by a timer. The timer was mainly implemented for the game environment to enhance the game factor.

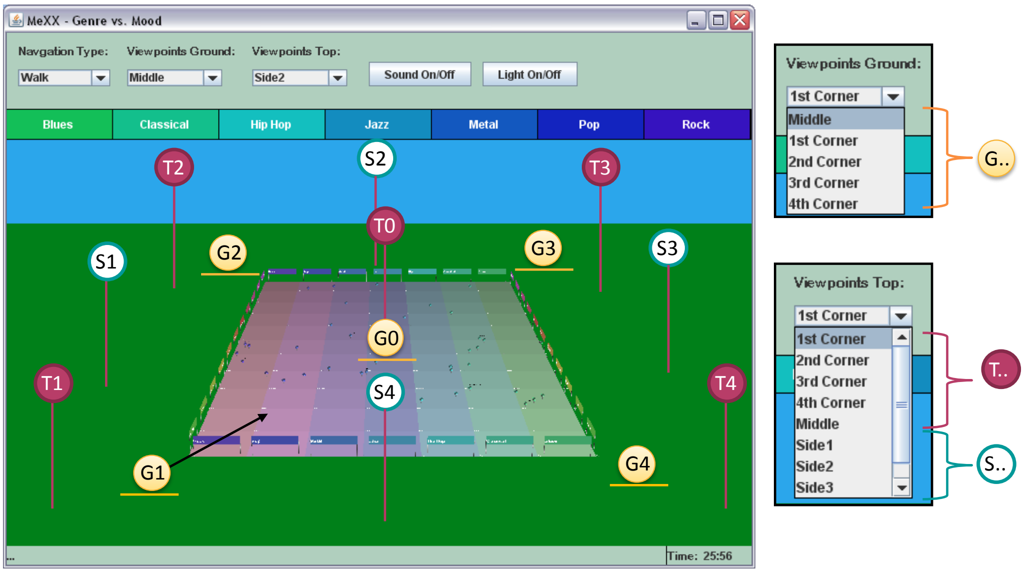


Figure-48: Viewpoint positions for teleporting

Game Layout

The game layout of the 3D world GUI is an extension to the visualization layout. Thus it contains all the features as mentioned for the first layout but has an additional panel, the so called song panel, and the “Show Result” button in the control panel; they are marked in Figure-49.

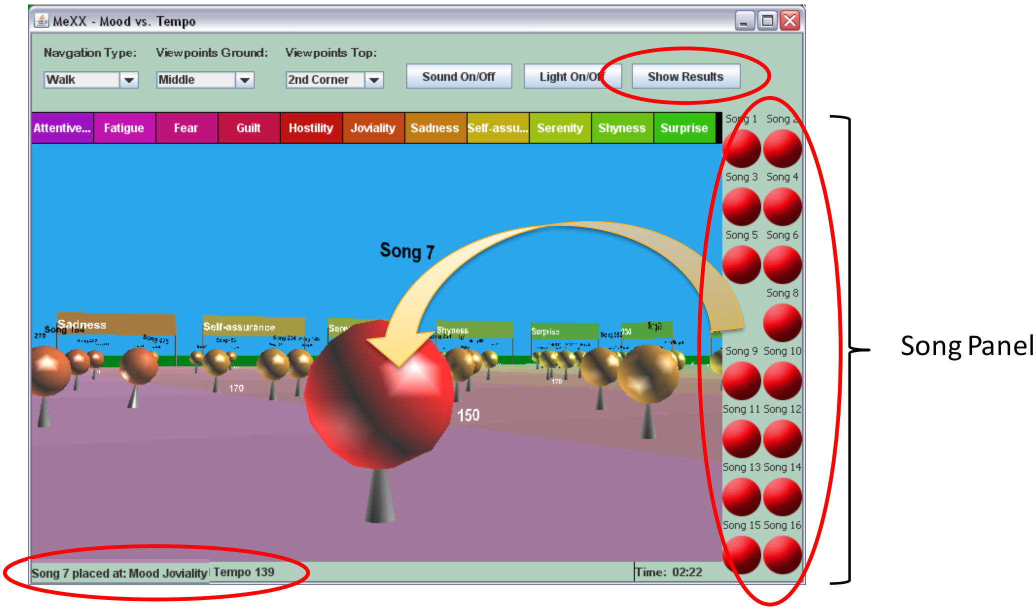


Figure-49: The 3D world GUI, game layout

The song panel contains the music pieces which need to be inserted into the 3D world by the user; these so called outer songs are represented through red spheres. To listen to them the left mouse buttons activates sound playback.

An insertion of an outer song into the world is possible by a click with the right mouse button. Through a small animated back step of the user's position the outer song appears in the 3D world and disappears in the song panel. The assigned feature values are then displayed on the output panel for control and confirmation purposes. Undo of an insertion is possible by a right mouse click at the particular outer song; the song vanishes in the 3D world and reappears in the song panel.

After all outer songs are inserted in the 3D world the "Show Result" button is enabled to show the result GUI which is depicted in Figure-50. For every outer song the title as well as the true solutions for the feature values, the obtained points and the elapsed time are displayed. Additionally the user can listen to the outer songs again in same way as in the song panel for an enhanced feedback and control.

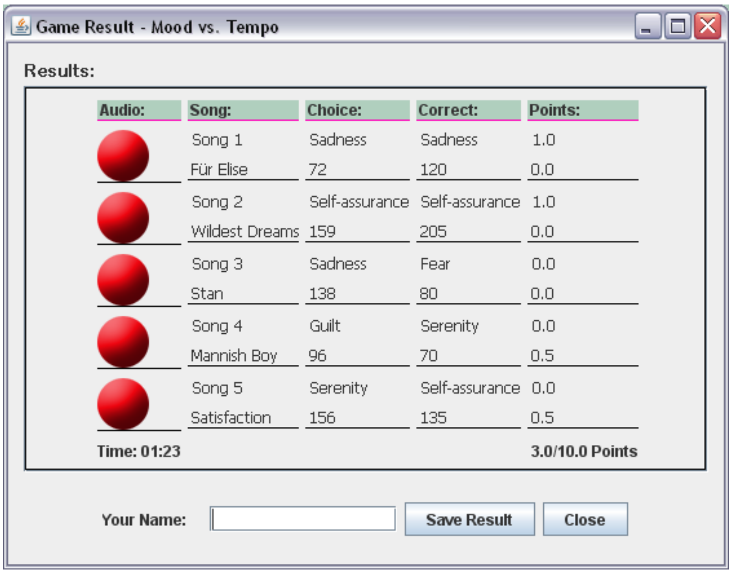


Figure-50: The result GUI

Per outer song 1 point is scheduled for each correct feature value, thus 2 points can be reached in total. In case of discrete feature values the score is increased by 1 point only if estimated and true feature values are identical. In case of numerical feature values, the estimated feature values are compared against two tolerance boundaries. If an outer song was assigned within the smaller boundary the score is increased by 1 point. Placement of a song within the larger boundary still raises the score for ½ points.

For a later examination or to recall the results, the user can optionally store the results in the text file `Results.txt` by typing his name and pushing the "Save Result" button. This file contains the game results for all played games and the actual result will be appended to this file. Additionally the player's name, actual date and time are also inserted. The file can then be analyzed by another system concerning the human's comprehension of the used features. In the next chapter application scenarios of MeXX are presented.

4.2 Application Scenarios

In this chapter eight scenarios using MeXX are presented and several screenshots are shown. First visualization scenarios of different music repositories are described; later applications in playing a game are depicted. Several music repositories were created to provide different data bases.

Scenario 1, Large Dataset

The first scenario visualizes a music repository that consists of 951 music pieces of 545 artists and 691 albums. The scenario is shown in Figure-51 whereas *genre* and *year* where chosen as axes features and *genre* has become additionally the color feature. The sphere layout represents the music pieces within the 3D world. The feature *year* is placed along the z-axis (left-up in the graphics) and ranges from 1970 to 2010. The feature *genre* is mapped on the x-axis (right-up in the graphics) and contains sections for `Classical`, `Country`, `Dance / Electronic`, `Hip-Hop`, `Jazz`, `Latin`, `R&B` and `Rock`. Using the third corner viewpoint as shown in Figure-51, the user quickly gets an overview of the music piece's distribution over the years and genres. The ground view of a modification of that scenario is depicted in Figure-52, in this case *year* was chosen as the color feature. When navigating through the world perception of 3D sound is possible.

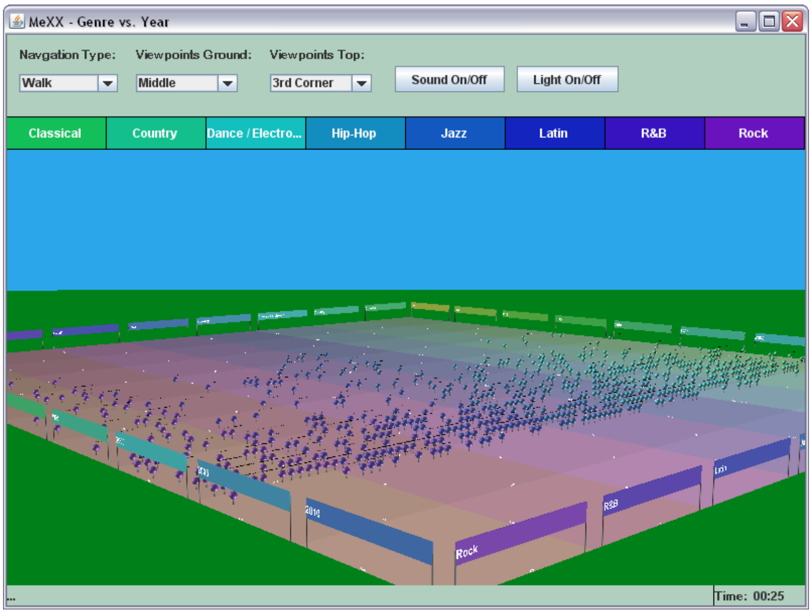


Figure-51: Scenario 1, large dataset, top view

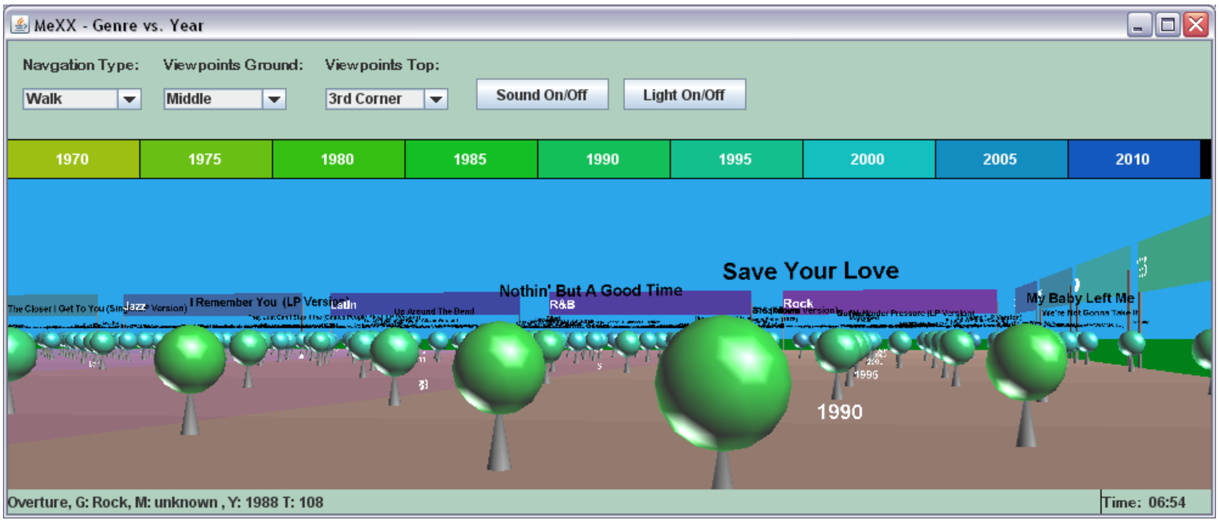


Figure-52: Scenario 1, large dataset, large ground view

Scenario 2, Database Filter

Within the second scenario the same music repository was used as in scenario 1, but in this case a filter was applied to the database. Only music pieces having the *genres* `Jazz`, `Latin` and `R&B` published between 1990 and 2010 are used. Thus the resulting 3D world consists of 339 music pieces, the album layout was chosen and *genre* was used as the color feature. Figure-53 shows a screenshot of the top view of the resulting 3D world whereas the *genre* is mapped the x-axis (left-right in the graphics) and *year* to the z-axis. In Figure-54 the night simulation was additionally switched on to force the fun factor. In Figure-55 a large ground view is shown with regular light condition.

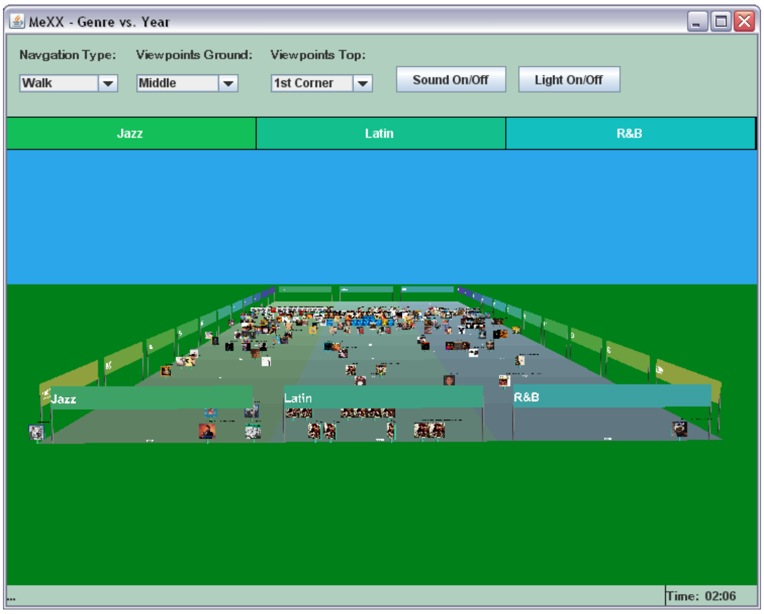


Figure-53: Scenario 2, database filter, top view

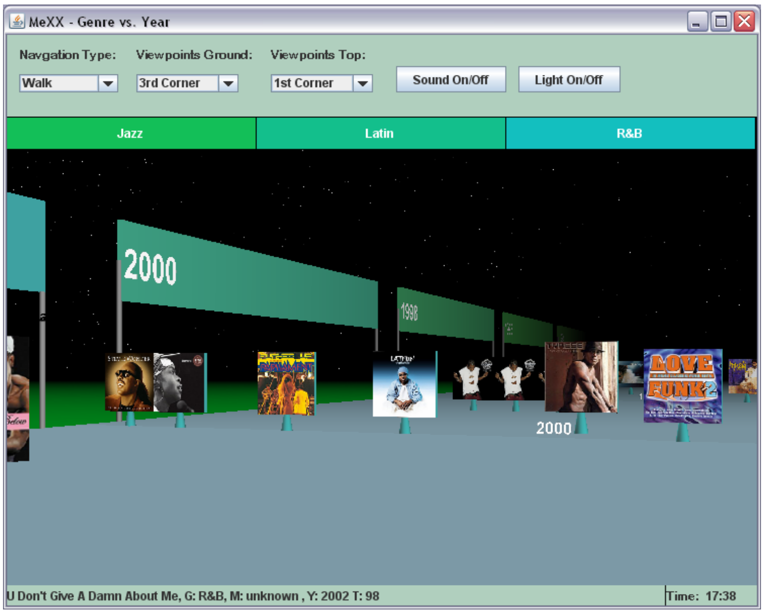


Figure-54: Scenario 2, database filter, ground view and night simulation

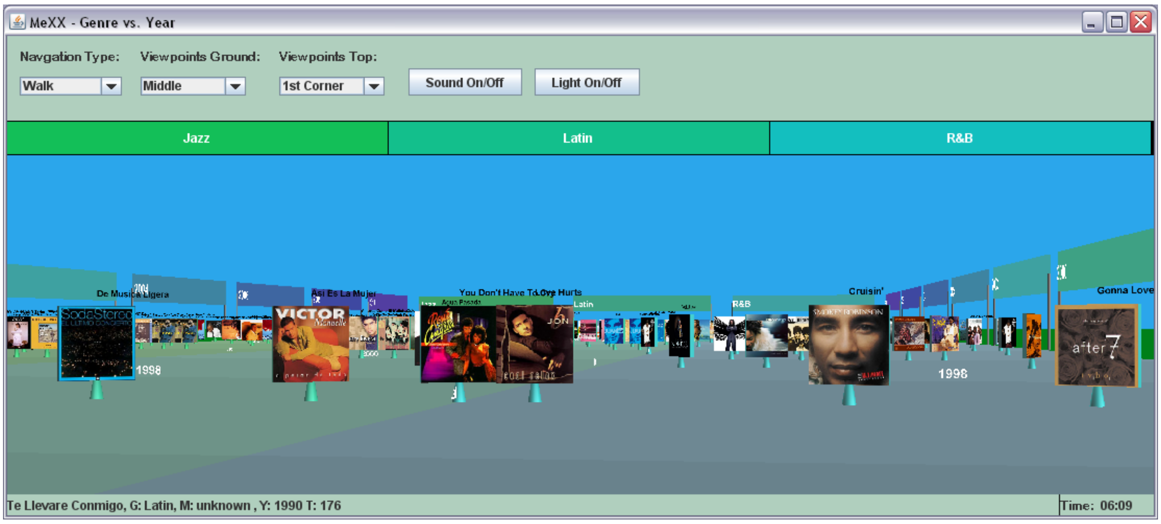


Figure-55: Scenario 2, database filter, large ground view

Scenario 3, Music History

A screenshot of the third scenario is shown in Figure-56. The underlying database consists of 38 music pieces, 20 artists and 24 albums published between 1700 and 2010. The database holds the *genres* `Blues`, `Classical`, `Hip Hop`, `Jazz`, `Metal`, `Rock` and `Pop`, and the *moods* `Fear`, `Hostility Guilt`, `Sadness`, `Joviality`, `Self-assurance`, `Attentiveness`, `Shyness`, `Fatigue`, `Serenity` and `Surprise`. The music pieces are arranged by the features *genre* and *year* represented through the sphere layout and color encoded by their *mood*. Additionally the corresponding albums and artists are placed within the 3D world since the music repository contains complete entries of their attributes. Within this scenario the user surfs in music history and perceives sound composed in different centuries.

Figure-56: Scenario 3, music history, ground view

Scenario 4, Pink Floyd

The fourth scenario presented in Figure-57 is a "homage" to the famous British rock band Pink Floyd. A database was created that holds 22 music pieces of 2 albums of the band which were categorized into the *genres* `Dance / Electronic` and `Rock` and hold *tempo* assignments ranging from 100 to 200 bpm. In the graphics the album layout was chosen to represent the music pieces which are arranged in terms of their *genre* and *tempo,* which additionally was chosen as the color feature. Navigating through the 3D world the user purely perceives the album artwork and sound of Pink Floyd.

Figure-57: Scenario 4, Pink Floyd, ground view

Scenario 5, Knocking On Heavens Door

The fifth scenario is shown in Figure-58. In this case a music repository was visualized that only contains interpretations of different artists of the popular song Knocking On Heavens Door by Bob Dylan. The database contains eight versions of the song and a complete entry for the corresponding albums and artists as they also show up in the 3D world. The *genres* `Rock`, `Reggae`, `Hardrock`, `Grunge` and `Blues` as well as the *moods* `Fear`, `Sadness` and `Serenity` were assigned to the music pieces. They are released between 1970 and 2000 and have a *tempo* between 60 and 80 bpm. In the screenshot they are arranged by *genre* and *year* as indicated by the top labels along the axis. The music pieces are represented by the sphere layout and color encoded by their *mood*. Figure-59 depicts the ground view of the 3D world where the user can navigate through to retrieve performance and sound changing and genre switching of that particular song in an entertaining way.

Figure-58: Scenario 5, Knocking On Heavens Door, top view

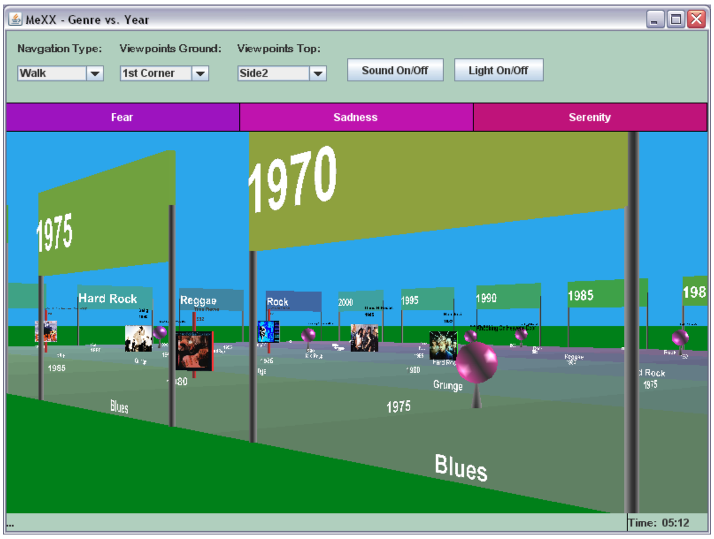


Figure-59: Scenario 5, Knocking On Heavens Door, ground view

Scenario 6, Poncho Sanchez (Game Option)

The sixth scenario was set up as a game as shown in Figure-60. A music repository that contains 22 music pieces of the American jazz musician Poncho Sanchez was visualized whereas 8 songs were chosen to be challenged on. The music pieces are arranged by their only *genre* `Jazz` and their *tempi* ranging from 90 to 200 bpm. In this case the user is only asked to place the 8 songs to their correct tempo sections. For the remaining music pieces in the 3D world the album layout was used and during game play the night simulation was switched on as a variation.

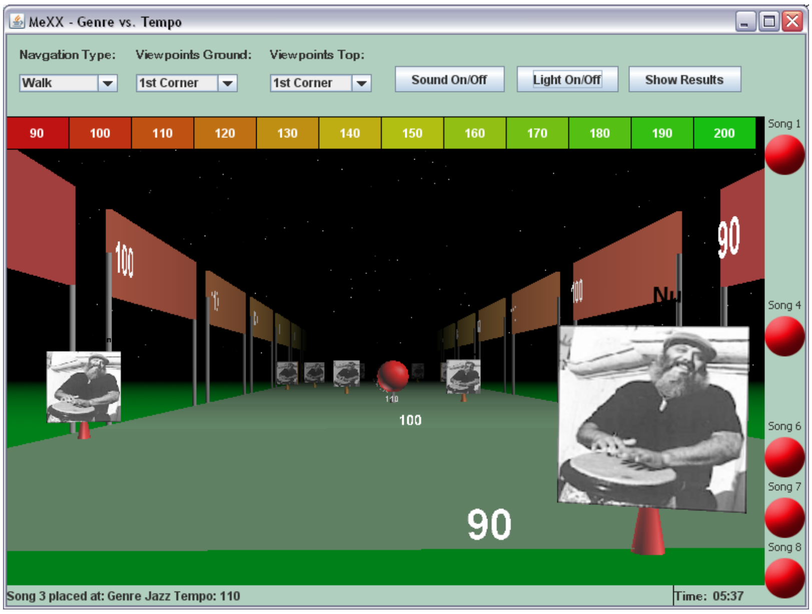


Figure-60: Scenario 6, Poncho Sanchez (game option), ground view and night simulation

Scenario 7, Music History (Filter and Game Option)

In the seventh scenario as shown in Figure-61, a game environment was established using the music repository mentioned in scenario 3. But in this case *mood* and *tempo* were selected as axes features and *mood* as the color feature. The album layout was used to represent the music pieces in the 3D world. Only the *moods* `Attentiveness`, `Fear`, `Guilt`, `Hostility`, `Joviality`, `Sadness`, `Self-assurance` and `Serenity` were chosen to be challenged on, the *tempi* range from 60 to 210 bpm. 10 outer songs were chosen to be challenged on their *tempo* and *mood*.

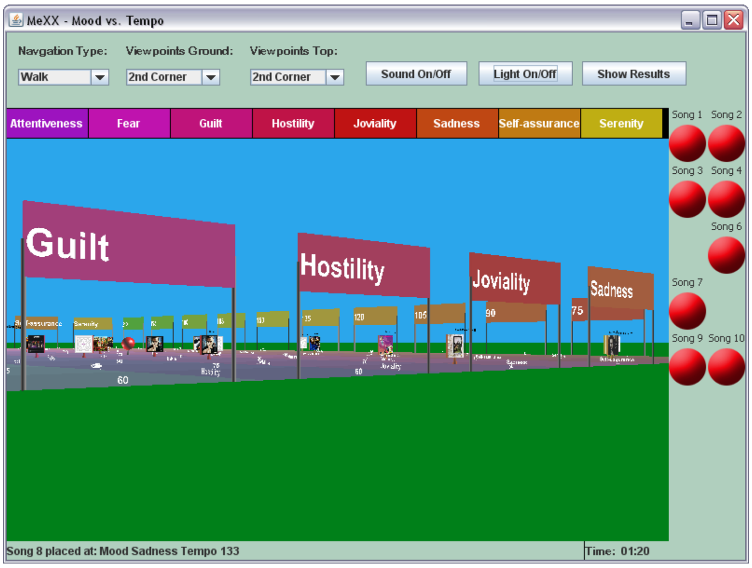


Figure-61: Scenario 7, music history (game option), ground view

Scenario 8, Large Dataset (Filter and Game Option)

The eighth scenario is shown in Figure-62 whereas a game environment was established. The music repository containing 951 music pieces was used whereas *genre* and *year* were selected to be mapped on the world axes. Additionally a filter was applied to limit the challenged feature values. Only music pieces that belong to the *genres* `R&B`, `Classical`, `Country` and `Dance / Electronic` and were published between 1990 and 2010 having a *tempo* between 130 to 210 bpm were chosen. *Tempo* was selected as the color feature and the album layout was used. The game was started with 20 outer songs. After inserting all songs in the 3D world the user gets a feedback including the solution, the obtained score and the elapsed time. The result can optionally be saved into the `Results.txt` file for further examination.

Figure-62: Scenario 8, large dataset (filter and game option), ground view

4.3 Summary

In this chapter MeXX was presented. First the Graphical User Interface (GUI) and the utilization of MeXX were explained in section 4.1. The music repositories are defined in the `DB.config` file whereas more than one database reference can be inserted. The start GUI consists of several options to configure the 3D world including the feature matrix, axes scaling, the database filter, the color- and appearance selection and the sound- and game options. Through the filter GUI a subset of music pieces out of a repository can be selected. The 3D world GUI is the actual visualization container of the 3D world. Depending on whether or not the game option has been chosen either the visualization layout or the game layout is established. The 3D world GUI consists of several options to manipulate the world including the selection of a navigation style and view point changes. Additionally sound and appearance options are available. Within the game environment of MeXX, a custom number of so called outer songs are separately displayed on the frame as red spheres, whereby listening to them is possible. After all outer songs are positioned into the world, the result GUI shows the solution and the obtained score. Optionally the game results can be appended to a text file.

In section 4.2 application scenarios for visualization and game playing tasks were presented with different music repositories created as appropriate databases. Browsing through music histories and establishing the 3D world as a “homage“ to particular artists and songs were presented. Additionally scenarios were created with complete or filtered music repositories arranged within the 3D world. Different axes and color features were used to customize the view. The sphere and album layout as well as the night simulation were activated for various impressions. The next chapter finally draws some conclusions and discusses future work.

5 Conclusion and Future Work

In this master thesis a virtual world was developed to explore music in terms of consumption and edutainment. The examination of state of the art Music Information Retrieval systems, Virtual Environments, Music-Based Games and Music Environment Systems yield to the design of a customizable 3-dimensional world with the options to visualize music repositories as well as to explore such repositories in an edutaining way. In such a 3D world music pieces are the central elements, they are visualized and arranged according to their particular features. Navigation within that world allows to visit particular music pieces, the concurrent playback of the corresponding audio files rendered as 3D sound was found to be a valuable feature of such an application.

For the implementation several tools including Sun's Project Wonderland, the Torque Game Engine and VRML/X3D were examined. Finally Java3D was chosen since it provides a collection of high-level constructs to create, render and manipulate 3D scenes. It can be extended or integrated by any Java application and additionally supports both, spatial and background playback of audio data. To establish such a 3D sound the Java3D *PointSound* class in combination with the JOALMixer based on the Java binding OpenAL project as audio device was used.

To provide music information databases were created that hold the information about the music pieces, artists and albums. The music pieces were chosen to be the central elements of the 3D world whereby the features *genre*, *mood*, *tempo* and *year* are used as candidates for their arrangement. Two of those features are mapped on the axes which span the world's floor and thus build the arrangement base. Optionally an additional feature can be used to color encode the visualized music pieces. Two different layouts, the sphere layout and the album layout control the appearance of the music pieces. In addition artists and albums can optionally be arranged in the world. For the database setup the MySQL Server was used, selective access to the data has been enabled by a filter.

The application is called MeXX and can be used to entertainingly deliver overviews about and insights into music. This includes the development of genres over the years or the artwork of a single artist through the 3D world. The game option of MeXX forces the user to correctly position a sub-set of music pieces in the world with the aim to score points. The purpose of the game is to challenge the users' sense and appreciation for music in an entertaining way. A night simulation was additionally implemented to increase the fun factor.

MeXX was tested with several music repositories holding up to a few 100 to 1.000 music pieces. Experiments showed that this is a proper size to render and display them, above that size long waiting periods and unexpected results may appear due to the rendering process of both, the visual and the aural objects.

Future work is required due to the number of rendered music pieces and number of concurrently played audio files. The automatic generation of level of detail nodes and mip-maps for textures are tasks to speed up the rendering performance of really large datasets. Development of a tool or the revision of the JOALMixer for a more controllable audio buffering to avoid unexpected results in rendering several hundreds of audio files seems to be necessary. In addition such a tool should provide the spatialized playback of (the popular) MP3 format.

A further approach is to enhance the dynamics of MeXX in terms of music features for the integration of arbitrary features, like rhythm or instrumentation of the music pieces. The mapping of the features to the axes is currently performed by a linear transform. In that case the sections may be sometimes overloaded whereas others remain almost empty. Mechanisms to automatically adjust the section sizes and to select and apply non-linear transform functions (e.g. logarithmic) probably enhance the appearance of the 3D world.

Actually the music pieces are arranged in the 3D world on a rectangular planar grid. Future work can be done to arrange the music pieces on arbitrary geometrical constructs like circles, spheres or spirals. More generally the use of the world's third dimension as an additional arrangement base will enrich the navigation possibilities and thus the user's feeling on flying through the space.

Currently three features of a music piece are used to be displayed; two are mapped to the axes and the third is color encoded. A special highlight could be to increase the number of features building the arrangement base of the virtual world by the implementation of a proper navigation style through a hyperspace. In such a multi-dimensional world the use of dynamic shapes in combination with transparency could be a first approach for a solution.

The game setup of MeXX was developed to train the user's appreciation for musical concepts. Detailed user studies required to evaluate the impact and pedagogic suitability of the system were beyond the scope of this thesis and need to be addressed in detailed follow-up studies.

An extension to assign music pieces to several discrete feature values is an interesting task to retrieve information about the humans' comprehension of music. This can be probably realized by a multi-user game environment which enables the particular users to assign the music pieces not only to one but to several feature values. An additionally task of such a game could be the completion or even the from scratch generation the of music pieces' metadata. The relationships between users and their feature assignments could probably create new classification schemes by use of modern machine learning techniques considering the user's attributes like age, gender, etc. tailor-made playlists based on customer needs can be established.

A sound agent can be developed to monitor music libraries and to extract features of the music pieces. These automatically generated features will be used to update a virtual environment as provided by MeXX with respect to new added music pieces.

A further vision is the implementation of a surround sound system in combination with virtual reality techniques like head mounted displays or CAVE's. Generally when thinking about music, modern 3D visualization and sound rendering techniques in combination with the possibilities the current state of virtual reality provides, there are virtually no restrictions in forming creative and artistic visions to experience music.

6 Bibliography

[1] **Apfelthaler, Kathrin. 2009.** *Potenziale des Virtuellen Lernraumes Second Life.* Master Thesis. Vienna, Austria : Vienna University of Technology, 2009.

[2] **Bainbridge, David, Cunningham, Sally Jo and Downie, Stephen, 2004.** Visual Collaging of Music in a Digital Library. *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004).* Barcelona, Spain, 2004, pp. 397-402.

[3] **Bainbridge, David, Nevill-Manning, Craig, Witten, Ian, Smith, Lloyd and McNab, Rodger, 1999.** Towards a Digital Library of Popular Music. *Proceedings of the 4th ACM International Conference on Digital Libraries (DL 1999).* California, United States, 1999, pp. 161-169.

[4] **Baum, Doris and Rauber, Andreas, 2006.** Emotional Descriptors for Map-Based Access to Music Libraries. *Proceedings of the 9th International Conference on Asian Digital Libraries (ICADL 2006).* Kyoto, Japan, 2006, pp. 370-379.

[5] **Baumann, Stephan, 2005.** A Music Library in the Palm of your Hand: A Series of Experiments on Interface Culture. *Proceedings of the Contactforum on Digital Libraries for Musical Audio.* Brussels, Belgium, 2005.

[6] **Brutzman, Don and Daly, Leonard. 2007.** *X3D: Extensible 3D Graphics for Web Authors.* San Francisco : Morgan Kaufmann Publishers, 2007.

[7] **Chen, Chaomei. 1999.** *Information Visualisation and Virtual Environments.* London : Springer, 1999.

[8] **Chen, Chin-Han, Weng, Ming-Fang, Jeng, Shyh-Kang, Chuang, Yung-Yu. 2008.** Emotion-Based Music Visualization Using Photos. *Lecture Notes in Computer Science Vol. 4903, Advances in Multimedia Modeling.* Berlin : Springer, 2008, pp. 358-368.

[9] **Croquet Consortium. 2010.** Croquet Consortium. [Online] 2010. [Cited: February 10, 2010.] http://www.opencroquet.org.

[10] **Davison, Andrew. 2005.** *Killer Game Programming in Java.* United States : O' Reilly Media, Inc., 2005.

[11] **Davison, Andrew. 2007.** *Pro Java 6 3D Game Developement - Java3D, JOGL, JINPUT, and* JOAL *APIs.* United States : Apress, 2007.

[12] **Dittenbach, Michael, Berger, Helmut, Genswaider, Ronald, Pesenhofer, Andreas, Rauber, Andreas, Lidy, Thomas and Merkl, Dieter. 2007.** Shaping 3D Multimedia Environments: The Media Square. *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR 2007).* Amsterdam, Netherlands, 2007, pp. 85-88.

[13] **Downie, Stephen J. 2003.** Music Information Retrieval. *Annual Review of Information Science and Technology, Vol. 37 (ARIST 37).* 2003, pp. 295-340.

[14] **Frank, Jakob, Lidy, Thomas, Peiszer, Ewald, Genswaider, Ronald and Rauber, Andreas. 2008.** Ambient Music Experience in Real and Virtual Worlds Using Audio Similarity. *Proceedings of*

the 1st ACM Workshop on Semantic Ambient Media (SAME 2008),. Vancouver, Canada, 2008, pp. 9-16.

[15] **Friberg, Johnny and Gärdenfors, Dan. 2004.** Audio Games: New Perspectives on Game Audio. *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology.* Singapore, 2004, pp. 148-154.

[16] **Gaitatzes, Athanasios, Christopoulos, Dimitrios and Roussou, Maria. 2001.** Reviving the Past: Cultural Heritage Meets Virtual Reality. *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage.* Glyfada, Greece, 2001, pp. 103-110.

[17] **GarageGames. 2010.** Torque Game Engine. [Online] 2010. [Cited: February 10, 2010.] http://www.torquepowered.com/.

[18] **Genswaider, Ronald. 2008.** *The Audio Square - A Collaborative Music Environment.* Master Thesis. Vienna, Austria : Vienna University of Technology, 2008.

[19] **Goto, Masataka and Goto, Takayuki. 2005.** Musicream: New Music Playback Interface for Streaming, Sticking, Sorting and Recalling Musical Pieces. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005).* London, United Kingdom, 2005, pp. 404-411.

[20] **Hase, Hans-Lothar. 1997.** *Dynamische Virtuelle Welten mit VRML 2.0.* Heidelberg : dpunkt, 1997.

[21] **Isaacson, Eric. 2005.** What you see is what you get: On Visualizing Music. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005).* London, United Kingdom, 2005, pp. 389-395.

[22] **Jung, Byungdae, Hwang, Jaein, Kim, Gerard Jounghyun, Lee, Ee-Taek and Kim, Hyunbin. 2000.** Incorporating Co-Presence in Distributed Virtual Environments. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology.* Seoul, Korea, 2000, pp. 206-211.

[23] **Kayali, Fares. 2008.** *Playing Music: Design, Theory and Practice of Music-Based Games.* Master Thesis. Vienna, Austria : Vienna University of Technology, 2008.

[24] **Knees, P., Schedl, M., Pohle, T. and Widmer, G. 2007.** Exploring Music Collections in Virtual Landscapes. *IEEE Multimedia Vol. 14 (3).* 2007, pp. 46-54.

[25] **Kolhoff, Philipp, Preuß, Jacqueline and Loviscach, Jörn. 2008.** Content-Based Icons for Music Files. *Computer & Graphics Vol. 32 (5).* 2008, pp. 550-560.

[26] **Lamere, Paul and Eck, Douglas. 2007.** Using 3D Visualizations to Explore and Discover Music. *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007).* Vienna, Austria, 2007, pp. 173-174.

[27] **Law, Edith L.M., von Ahn, Luis, Dannenberg, Roger B. and Crawford, Mike. 2007.** TagATune: A Game for Music and Sound Annotation. *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007).* Vienna, Austria, 2007, pp. 361-364.

[28] **Lee, Jin Ha and Downie, Stephen J. 2004.** Survey of Music Information Needs, Uses and Seeking Behaviors: Preliminary Findings. *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004).* Barcelona, Spain, 2004, pp. 441-446.

[29] **Leitich, Stefan and Topf, Martin. 2007.** Globe of Music - Music Library Visualization Using GeoSOM. *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007).* Vienna, Austria, 2007, pp. 167-171.

[30] **Lepouras, Georg and Vassilakis, Costas. 2004.** Virtual Museums for All: Emplying Game Technology for Edutainment. *Virtual Reality Vol. 8 (2).* 2004, pp. 96-106.

[31] **Linden Research. Inc. 2010.** Second Life. [Online] 2010. [Cited: February 10, 2010.] http://secondlife.com.

[32] **Lübbers, Dominik. 2005.** SoniXplorer: Combining Visualization and Auralization for Content-Based Exploration of Music Collections. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005).* London, United Kingdom, 2005, pp. 590-593.

[33] **Mandel, Michael I. and Ellis, Daniel P.W. 2007.** A Web-based Game for Collecting Music Metadata. *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007).* Vienna, Austria, 2007, pp. 365-366.

[34] **Marschner, Mark Oliver. 2008.** *Foundation for Music-Based Games.* Master Thesis. Vienna, Austria : Vienna University of Technology, 2008.

[35] **Mateevitsi, Victor, Sfakianos, Michael, Lepouras, George and Vassilakis, Costas. 2008.** A Game-Engine Based Virtual Museum Authoring and Presentation System. *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts.* Athens, Greece, 2008, pp. 451-457.

[36] **Mayer, Rudolf, Lidy, Thomas and Rauber, Andreas. 2006.** The Map of Mozart. *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006).* Victoria, Canada, 2006.

[37] **Mazzoleni, Pietro, Bertino, Elisa, Ferrari, Elena and Valtolina, Stefano. 2004.** CiVeDi: A Customized Virtual Environment for Database Interaction. *ACM SIGMOD Record, Vol.33 (3).* 2004, pp. 15-20 .

[38] **Milgram, Paul, Takemura, Haruo, Utsumi, Akira and Kishino, Fumio. 1994.** Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum. *SPIE Vol. 2351, Telemanipulator and Telepresence Technologies.* 1994, pp. 282-292.

[39] **Müller, Meinard. 2007.** *Information Retrieval for Music and Motion.* Berlin, Germany : Springer, 2007.

[40] **Neumayer, Robert, Dittenbach, Michael and Rauber, Andreas. 2005.** PlaySOM and PocketSOM Player: Alternative Interfaces to Large Music Collections. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005).* Victoria, Canada, 2005, pp. 618-623.

[41] **Orio, Nicola. 2006.** Music Retrieval: A Tutorial and Review. *Foundations and Trends in Information Retrieval Vol. 1 (1).* 2006, pp. 1-90.

[42] **Pampalk, Elias and Goto, Masataka. 2006.** Music Rainbow: A New User Interface to Discover Artists Using Audio-Based Similarity and Web-Based Labeling. *Proceedings of the 7th International Conference of Music Information Retrieval (ISMIR 2006).* Victoria, Canada, 2006, pp. 367-370.

[43] **Pampalk, Elias. 2003.** Islands of Music - Analysis, Organization, and Visualization of Music Archives. *Journal of the Austrian Society for Artificial Intelligence (ÖGAI), Vol. 22 (4).* 2003, pp. 20-23.

[44] **Pichlmair, Martin and Kayali, Fares. 2007.** Levels of Sound: On the Principles of Interactivity in Music Video Games. *Situated Play: Proceedings of the Third International Conference of the Digital Games Research Association (DiGRA 2007).* Tokyo, Japan, 2007, pp. 424-430.

[45] **Stavness, Ian, Gluck, Jennifer, Vilhan, Leah and Fels, Sidney. 2005.** The MUSICtable: A Map-Based Ubiquitous System for Social Interactio with a Digital Music Collection. *Proceedings of the 4th International Conference on Entertainment Computing (ICEC 2005).* Sanda, Japan, 2005, pp. 291-302.

[46] **Sun Microsystems, Inc. 2010.** Java3D Project. [Online] Sun Microsystems, Inc., 2010. [Cited: February 10, 2010.] https://java3d.dev.java.net/.

[47] **Sun Microsystems, Inc. 2010.** Project Wonderland. [Online] Sun Microsystems, Inc., 2010. [Cited: February 4, 2010.] http://www.projectwonderland.com/.

[48] **Thalmann, Florian. 2006.** Jam Tomorrow: Collaborative Music Generation in Croquet Using OpenAL. *Proceedings of the 4th International Conference on Creating, Connecting and Collaborating Through Computing (C5 2006).* California, United States, 2006, pp. 79-87.

[49] **The American Heritage Dictionary. 2004**. Dictionary.com [Online] 4th Edition, Houghton Mifflin Company, 2004. [Cited: February 17, 2010.] http://dictionary.reference.com/browse/edutainment

[50] **Torrens, Marc, Hertzog, Patrick and Arcos, Josep-Lluís. 2004.** Visualizing and Exploring Personal Music Libraries. *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2005).* Barcelona, Spain, 2004, pp. 10-14.

[51] **Tsichritzis, Dennis and Gibbs, Simon. 1991.** Virtual Museums and Virtual Reality. *Proceedings of the 1st International Conference on Hypermedia and Interactivity in Museums (ICHIM 1991).* Pittsburgh, United States, 1991, pp. 17-25.

[52] **Turnbull, Douglas, Liu, Ruoran, Barrington, Luke and Lanckriet, Gert. 2007.** A Game-based Approach for Collecting Semantic Annotations of Music. *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007).* Vienna, Austria, 2007, pp. 535-538.

[53] **Typke, Rainer, Wiering, Frans and Veltkamp, Remco. 2005.** A Survey of Music Information Retrieval Systems. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005).* London, United Kingdom, 2005, pp. 153-160.

[54] **van Gulik, Rob, Vignoli, Fabio and van de Wetering, Huub. 2004.** Mapping Music in the Palm of Your Hand, Explore and Discover Your Collection. *Proceedings of 5th International Conference on Music Information Retrieval (ISMIR 2004).* Barcelona, Spain, 2004.

[55] **Vavrille, Frederic. 2005.** Liveplasma. [Online] 2005. [Cited: February 10, 2010.] http://www.liveplasma.com/.

[56] **Vince, John. 1998.** *Virtual Reality: How to Unserstand the Techniques and Potentials of Virtual Reality.* Great Britan : Springer, 1998.

[57] **Wattenberg, Martin. 2001.** The Shape of Song. [Online] 2001. [Cited: February 10, 2010.] http://www.turbulence.org/Works/song/.

[58] **Web3D Consortium. 2010.** VRML/X3D Specification. [Online] 2010. [Cited: February 10, 2010.] http://www.web3d.org/.

[59] **Yilmaz, Erdal, Cetin, Yasemin, Erdem, Cigdem, Erdem, Tanju and Özkan, Mehmet. 2006.** Music Driven Real-Time 3D Concert Simulation. *Lecture Notes in Computer Science Vol. 4105, Multimedia Content Representation, Classification and Security.* Berlin : Springer, 2006, pp. 379-386.

[60] **Yoshii, Kazuyoshi and Goto, Masataka. 2008.** Music Thumbnailer: Visualizing Musical Pieces in Thumbnail Images Based on Acoustic Features. *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008).* Philadelphia, United States, 2008, pp. 211-216.