UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

Video Browsing and Soundtrack Labelling based on Human Computation

Jorge Miguel Correia Antunes Gomes

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Engenharia de Software

2013

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

Video Browsing and Soundtrack Labelling based on Human Computation

Jorge Miguel Correia Antunes Gomes

DISSERTAÇÃO

Supervised by Prof. Dr. Maria Teresa Caeiro Chambel

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Engenharia de Software

2013

Acknowledgements

I would love to express my sincere gratitude to everyone who supported me in the last months, especially to my supervisor Prof. Dr. Teresa Chambel, and Prof. Dr. Thibault Langlois, the leader of the VIRUS project, for being available and the continuous support, the long discussions filled with the constructive criticism, the interesting ideas and all the help and collaboration provided, which made this work feasible and enjoyable.

Likewise, to my close family members - Beatriz Gomes, Fernando Gomes, Amália Moura, Daniela Gomes, Pedro Gomes, Inês Gomes - for the support and caring during all these years in good and bad moments, without them I would not have this opportunity.

I would like to deeply thank all my friends: To Fernando Fernandes for his amazing friendship, great advices, the shared adventures and initiating (with) me in geocaching. To Alexander Fernandes for being such an amazing friend, the shared moments, projects and games, and staying with me through difficult moments. To Tiago Senhorinho for his great company, the shared moments, conversations and adventures, the honesty, being an amusing person, and especially for being the oldest friend I ever had. To Camila Novais for her good pieces of advice, the motivation and engaging conversations. To Edite Fernandes for her support, friendship, cheerful company and shared adventures, for pushing me to new places and opening my world to new experiences. To Tamzyn Do Couto (and Zara) for her friendship, inspiration and positivism, for being supportive and lovely to offer me (extreme) body conditioning for the next time I visit a McDonald's, for her honesty, her extensive philosophical conversations and arguments, and by helping me improving my English. To Oleksandr Malichevskyy for the shared friendship and adventures, for the LaTEX tips and proving that coffee is still better than energy drinks, including his recent discovery of Rodeo. To all people at Rota Jovem for the warm welcome and supporting my ideas, the developed activities, and for the opportunities provided, especially to Hugo Matos for the natural zeal and passion he puts in his work, and by withstanding me in the train. To Tiago Inocêncio, Paulo Ribeiro and João Oliveira for sharing good moments and epic lunch conversations. To César Santos and Ana Narciso for sharing good music and moments. To Silvana Silva for all the amusing cat images and videos we shared. To Bruno Neves for the many coffees talks and the mutual support. To João Gonçalves, André Pires, Rui Rodrigues, Telma Martins, Patrícia Silva, João Rico, Claudia, André Pantaleão and Bruno Russo for the company, the amusing nights out and conversations at

the coffee shop. To Filipe Carvalho for the great friendship, the experiences shared and the great company provided, especially during my vacations. Not forgetting of course about Alexandra Pinto, André Rosa, Custodia Carvalho, Eduardo Pais, Gabriel Carvalho, Inês Vasques, Jason Heron, João Tavares, José Pedrosa, Kleomar Almeida, Meryeli Macedo, Miguel Melo, Nuno Cardoso, Raul Pinto and Ricardo Wilhelm. I would detail this list even more but no words would be enough to express how much everyone contributed to my personal growth and, directly or indirectly, affected my work done throughout this year. My best wishes to everybody I mentioned before!

My humble gratitude to all my colleagues, especially to Eduardo Duarte for teaching the basics of his audio similarity tool, to Nuno Gil by sharing his work and reports, to Ana Jorge for sharing ideas and the opportunity to participate in her project. My thanks to all volunteers who participated in SoundsLike evaluations, especially, Vinicius Cogo for the time spent giving tons of feedback, suggestions and cool ideas for my work.

I am graceful for the support provided by FCT: Fundação para Ciência e Tecnologia through a research grant in the VIRUS project (PTDC/EIA-EIA/101012/2008). My acknowledgements to the Department of Informatics, Faculty of Sciences, University of Lisbon, and the HCIM group at LaSIGE, for providing all the necessary conditions for the development and writing of this thesis.

To finish, my regards to everyone who contributed to the open-source libraries and software I used to develop my work and produce this document, to the Facebook for the constant procrastination temptation, to the NSA[1] for spying on me and everyone else, and to you, yes you!, the reader, one of the reasons which I have created this magnificent document.

[1]Nacional Security Agency

To everyone who supported me, through good or difficult moments!

Resumo

A rápida expansão dos meios de comunicação permitiu a criação de enormes e complexas colecções multimédia acessíveis através da Internet e media sociais. Essas colecções exigem novos mecanismos de busca, que irão beneficiar de novas técnicas de classificação e análise automática de conteúdo de áudio e vídeo, e de relações entre os documentos multimédia e os seus conteúdos.

O vídeo é um meio muito rico, combinando imagem e som e proporcionando assim uma enorme quantidade de informação e uma excelente plataforma para a criatividade ser expressa e explorada. A riqueza do vídeo e multimédia permite a extracção de propriedades interessantes que podem ser úteis em sistemas de pesquisa exploratória.

No entanto, toda esta riqueza que torna tão interessantes os espaços de informação de áudio e vídeo, traz consigo uma complexidade com que é difícil lidar. Eventos sonoros, emoções expressas e sentidas, estados de espírito, cores e ritmo sonoro são exemplos de propriedades multimédia interessantes a serem exploradas.

Alguns investigadores apontaram a importância para o desenvolvimento de métodos para extrair características interessantes e significativas em vídeos para efectivamente resumi-los e indexá-los ao nível das legendas, imagem e áudio.

As tarefas de análise de áudio e vídeo são consideradas como árduas e complexas. As abordagens mais comuns baseiam-se em modelos estatísticos, que exigem a construção de conjuntos de dados classificados compostos por amostras de vídeo e áudio. A construção de um conjunto de dados exigiria horas incontáveis, de escuta e de classificação manual - isto é muitas vezes formulado como o problema de "arranque a frio" ("cold-start problem") na aquisição de dados.

A colecta de informações detalhadas sobre o conteúdo multimédia poderá melhorar os sistemas automáticos de extracção de informação e facilitar a prospecção de dados, para extrair relações úteis para as indústrias de multimédia, melhorar sistemas de recomendação baseados em conteúdos, publicidade contextual e "personalized retargeting". Para tal, é necessária a exploração de novos métodos para a classificação de grandes quantidades de vídeos e áudio, com vista a solucionar o problema de "arranque a frio" da aquisição de dados, e providenciar um modo para a partilha de resultados e conjunto de dados com a comunidade científica.

Entretanto, é também importante a exploração de formas de visualização de informação relativas a vídeo e filmes, ao nível de cada um e ao nível do espaços de filmes, incluindo a representação temporal quer dentro dos filmes, que contém muita informação ao longo do tempo da sua duração, quer ao nível do tempo de lançamento ou visionamento de séries e filmes.

Hoje em dia, a importância da visualização de dados está a aumentar cada vez mais. Técnicas de visualização podem ser a opção mais forte para transmitir e expressar ideias a outras pessoas recorrendo a gráficos, diagramas e animações. No entanto, isto exigirá conhecimentos sobre a linguagem da comunicação visual, que envolvem semânticas e sintaxes semelhantes à linguagem verbal. Estas técnicas podem ajudar a lidar com a complexidade de dados e explorar modos avançados e eficazes para transmitir informações provenientes de espaços de informação.

Este projecto pretende contribuir na área de visualização e recuperação de informações de áudio e vídeo ao permitir aos utilizadores aperceberem-se e procurarem por certas propriedades multimédia e abordar o problema de "arranque a frio" ("cold-start problem"). A solução passa pela criação de novas abordagens que dependem de mecanismos interactivos de *crowdsourcing* e computação baseada em humanos que irão recorrer a elementos de jogos para motivar os utilizadores a contribuir para resolução do problema de "arranque a frio" na classificação de conteúdos.

Crowdsourcing aqui significa confiar nas contribuições provenientes de um enorme grupo de pessoas, especialmente de uma comunidade on-line ou redes sociais, onde tarefas possam ser concluídas por diversas pessoas. Neste contexto, pretende-se classificar documentos multimédia e procurar um consenso geral para obter informações relevantes que descrevam com precisão estes mesmos documentos em meta-dados sugeridos pelos utilizadores.

Isto irá criar bases de dados que poderão ser compartilhados e reutilizados pela comunidade científica, com vista a serem utilizados em modelos estatísticos que suportam a extracção automática de informação e prospecção de dados referidas anteriormente.

Este trabalho está relacionado com outros projectos, onde as características de vídeo são extraídos por meio de processamento do áudio (tiro, gritos, risos, humor música, etc), análise de legendas (usando a interpretação semântica e análise de sentimento relativa a emoções expressas) e monitorização das emoções dos espectadores através de dados biométricos (frequência cardíaca, a respiração, a resposta galvânica da pele, etc), ou reconhecimento visual de expressões faciais.

O trabalho relatado por esta dissertação foca-se nas dimensões interactivas para visualização, acesso e classificação de conteúdos em filmes, visualizações interactivas para espaços de filmes e para a representação de segmentos de áudio semelhantes, com base no teu conteúdo, e permitir uma navegação contextualizada de filmes e a classificação interactiva de conteúdos no SoundsLike. Adoptando uma abordagem de Computação ba-

seada em humanos, SoundsLike é um Jogos com um Propósito ("Game With A Purpose") que tem dois objectivos em mente: 1) utilizar elementos de jogos para entreter e motivar o utilizador na navegação e classificação de vídeos e filmes; e 2) utilizar esta interacção para recolher informação e melhorar técnicas de análise de áudio baseado em conteúdo, recorrendo a paradigmas de *crowdsourcing* para obter consensos sobre a relevância de dados recolhidos e pontuar correctamente cada contribuição. SoundsLike está integrado no MovieClouds, uma aplicação Web interactiva desenhada para aceder, explorar e visualizar filmes baseada na informação fornecida em diferentes perspectivas do seu conteúdo. Esta abordagem de classificação poderá posteriormente ser estendida para outros tipos de conteúdos, não se limitando somente à componente de áudio.

Palavras-chave: Navegação Interactiva, Áudio, Música, Banda Sonora, Vídeo, Filmes, Etiquetagem, Computação Baseada em Humanos, Jogo Com Um Propósito, Gamificação, Entretenimento, Motivação, MovieClouds, SoundsLike.

Abstract

Video and audio are becoming dominant media in our lives. In a time when we witness the convergence of media, it is pertinent the creation of new and richer ways for content-based access and visualization of videos. Collections of video require new search mechanisms which will benefit from new classification techniques and automatic analysis of video and audio content and relationships among them.

Video is a very rich medium combining image and sound, thus providing huge amounts of information. However, this richness that makes video and audio based information spaces so interesting comes with a challenging complexity to handle.

The exploration of new visualization methods related to video and movies at the movie space level down to the each movie itself, including representations along time of the videos' content or the time of their releases or viewing, allow pattern and trend analysis and movie browsing.

The exploration of new visualization methods may enhance video and movies perception and navigation at the movie space and the individual movie levels. Representations along time of the videos' content in their different perspectives (sound, subtitles, image, etc.) or the time of their releases or viewing, allow identifying and analysing use and content patterns and relations, for a richer understanding and access in movie browsing.

Game elements, in turn, can help in this often challenging process, e.g. in the audio, to obtain user feedback to improve the efficacy of classification, while maintaining or improving the entertaining quality of the user experience.

This dissertation's project aims to improve the area of visualization and information retrieval of audio and video, by adopting a Human Computation approach through a Game With A Purpose to entertain and engage users in movies soundtrack browsing and labelling to collect data that also improve our content-based sound classification techniques. SoundsLike is integrated in MovieClouds, an interactive web application designed to access, explore and visualize movies based on the information conveyed in the different tracks or perspectives of its content.

Keywords: Interactive Browsing, Audio, Music, Sountrack, Video, Movies, Labelling, Tagging, Human Computation, Game With A Purpose, Gamification, Entertainment, Engagement, User Experience, MovieClouds, SoundsLike.

Contents

List of Figures

List of Tables

Abbreviations

ACID	Atomicity Consistency Isolation Durability
AJAX	Asynchronous Javascript And XML
API	Application Programming Interface
CSS	Cascade Style Sheet
D3	Data-Driven Documents
FCUL	Faculdade de Ciências da Universidade de Lisboa
GUI	Graphical User Interface
GWAP	Game With A Purpose
HCIM	Human-Computer Interaction and Multimedia Research Team
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
LaSIGE	Large-Scale Informatics System Laboratory
LOC	Lines Of Code
MVC	Model-View-Controller
PDO	PHP Data Object
PEI	Projecto de Engenharia Informática (Computer Science Engineering Project)
RDMS	Relational Database Management System
PHP	PHP: Hypertext Preprocessor
REST	Representational State Transfer
RPC	Remote Procedure Call
SQL	Structured Query Language
SLOC	Software Lines Of Code
SVG	Scalable Vector Graphics
UML	Unified Modelling Language
URI	Uniform Resource Identifier
USE	Usefulness, Satisfaction and Ease of Use
VIRUS	Video Information Retrieval Using Subtitles
VML	Vector Markup Language
W3C	World Wide Web Consortium
WWW	World Wide Web

Chapter 1

Introduction

1.1 Motivation

Video and audio have a strong presence in human life, being a massive source of information. The fast expansion of media has enabled the creation of huge and complex multimedia collections accessible over the internet and social media. These collections require new search mechanisms which will benefit from new classification techniques and automatic analysis of video and audio content and relationships among them.

Video is a very rich medium combining image and sound, thus providing huge amounts of information and excellent platform for creativity to be expressed and explored [42]. The richness of video and multimedia enables the extraction of interesting properties that may be useful in exploratory search systems.

However, all the richness that makes video and audio based information spaces so interesting, inside each video or audio and outside in the way they relate to each other, comes with a challenging complexity to handle. Sound events, felt and expressed emotions, moods in audio, and colours are examples of interesting multimedia properties to be explored, due their usefulness to describe video contents in different perspectives. To extract such meaningful information from videos, mechanisms are required for the extraction of information from huge collections of videos.

Some researchers [10, 23] pointed out the importance of the development of methods to extract interesting and meaningful features in video to effectively summarize and index them at the level of subtitles, audio and video image.

But, video and audio analysis are considered complex and arduous tasks, and most formulated approaches are based on statistical models which require the construction of classified datasets composed by video and audio samples. The building of our own dataset would require many hours of listening and manual classification - this is often formulated as the "cold start" problem in data acquisition.

Gathering detailed information about multimedia content may improve automatic information retrieval systems and facilitate data mining over the extracted data, to extract

relationships useful for multimedia industries to improve content-driven recommendation systems, contextual advertisement and personalized retargeting. Therefore, the exploration for new ways to classify large amounts of videos and audio is necessary, to help towards the resolution of the cold start problem in data acquisition, and provide a way for sharing the results and dataset with the scientific community.

Meanwhile, it is also important the exploration of ways of visualizing the information related to video and movies, at the level of each movie and at the movies space level, including representation of time for release of shows and movies or the time when viewers watched the movies in different seasons, allowing trend analysis and pattern prediction.

Nowadays, the importance of data visualization in many areas is increasing. Visualization techniques can be the strongest option for expressing ideas to other people with graphs, diagrams and animation.These techniques can help to deal with the data complexity and explore enhanced and effective ways to transmit information from information spaces [16].

1.2 Objectives

This project aspires to contribute to the area of visualization and information retrieval of audio and video, and to enable users to search for certain properties and solve the cold start problem. The solution involves creating new approaches that rely on interactive crowd-sourcing mechanisms and human computation through game elements to compel users to contribute to solve the cold start problem and helping in the content classification.

Here, crowdsourcing means relying on contributions from a huge group of people, especially an online community or social networks, in a way that tasks could be completed by multiple persons. In this environment, it is intended to classify multimedia documents and resort to general consensus to obtain relevant information that describes with precision the same documents. This consensus boils down to metadata matching suggested by users. A larger number of people wrapped around the classification of a multimedia element may have higher chance to unveil a good consensus about a metadata group, thus achieving a higher significance for the same metadata.

This will create databases that could be shared and reused by the scientific community, to be used in statistical models supporting the automatic information retrieval and data mining aiming to extract relationships useful for the cinematographic and discography industries (i.e. to improve content-driven recommendation systems, contextual advertisement and personalized retargeting).

Further exploration of others aspects includes usual representation of time inside the movies and along the different perspectives of their content.

The work reported in this dissertation focuses on interactive dimensions for visualization, access and content classification for movies, the representation of similarity of audio

excerpts, based on content, and enabling the contextualized browsing of movies and TV series, act as a base for interactive content classification and browsing in SoundsLike. Adopting an approach based on human computation, SoundsLike is a Game With A Purpose (GWAP) that pursues two goals: 1) use game elements to engage the user in movie browsing and audio content classification, and 2) use this interaction to collect data and improve our content-based sound analysis techniques, using crowd-sourcing paradigms to achieve consensus on the relevance of collected data and reward users properly for the "significance" of their contribution. SoundsLike is integrated in MovieClouds [20], an interactive web application designed to access, explore and visualize movies based on the information conveyed in the different tracks or perspectives of their content.

1.3 Context

This master dissertation is elaborated as a PEI[1] integrated in the Computer Science Engineering Master course of FCUL[2] and was developed around the VIRUS "Video Information Retrieval Using Subtitles" (Langlois et al., 2010) research project, in the context of the HCIM group at LaSIGE, FCUL.

The dissertation focused on the development and testing of interfaces for new features that includes access and classification of audio segments according with its content to be integrated in MovieClouds - an interactive web application designed to access, explore and visualize movies based o the information conveyed in the different tracks or perspectives of their content -, motivating user contributions through the use of game elements together with crowdsourcing paradigms. It also explored new ways for interactive visualization of information and films along time.

It is linked with other master and PhD thesis where MovieClouds (N. Gil, N. Silva et al.) [20], iFelt (E. Oliveira, P. Martins, T. Chambel) [51] and ColoursInMotion (J. Martilho, T. Chambel) [42] where first designed and created, related to the retrieval of video characteristics through audio processing (shooting, screams, laughs, music mood, etc.), subtitles analysis (using semantics interpretation, including expressed emotions and feelings) and monitoring of viewer emotions through biometric data (heart rate, reactions, respiration, galvanic skin response, etc.) or recognition of facial expressions.

[1] *Projecto de Engenharia Informática* (Computer Science Engineering Project)
[2] Faculdade de Ciências da Universidade de Lisboa

1.4 Contributions

The main contributions of this project are the following:

- Characterization of the state of art;
- Representation and access to similar audio track excerpts obtained from movies;
- Classification of audio tracks using an application developed from a previous dissertation under the same project;
- Design and development of SoundsLike, an interactive Web Application for movie soundtrack audio classification integrated and deployed as a part of the MovieClouds prototype (the front-end and back-end composed by a middleware Web-Service and data storage).
- Design and Implementation of interactive and animated 2D visualizations to overviews and allows the browsing and watching of the movies' content through a wheel visualization with diverse range of tracks that represent different perspectives of the movies contents.
- Evaluation of the SoundsLike prototype through usability tests with users.

It is also important to note the contribution through three scientific papers for international conferences:

- Gomes, J.M.A., Chambel, T. and Langlois, T., 2013. SoundsLike: Movies Soundtrack Browsing and Labeling Based on Relevance Feedback and Gamification. In Proceedings of the 11th European conference on Interactive TV and video. Como, Italy, June 24-26: ACM, pp. 59–62.
- Gomes, J.M.A., Chambel, T. and Langlois, T., 2013. Engaging Users in Audio Labelling as a Movie Browsing Game With a Purpose. In 10th International Conference On Advances In Computer Entertainment Technology. Bad Boekelo, Enschede, Netherlands, November 12-15: Springer, Netherlands, 12 pages.
- Gomes, J.M.A., Chambel, T. and Langlois, T., 2014. A Video Browsing Interface for Collecting Sound Labels using Human Computation in SoundsLike. In 9th International Conference on Computer Graphics Theory and Applications. Lisbon, Portugal, January 5-8: SCITEPRESS Digital Library, 8 pages.

I also participated in the Faculty Open Day[3] on April 11, 2013 as student volunteer, presenting the to groups of high school students the developed work on interactive movie classification in SoundsLike and MovieClouds (Interactive Movie Classification: MovieClouds and SoundsLike[4], Jorge M. A. Gomes, Teresa Chambel, Thibault Langois). The open day was organized by the Faculty of Sciences, University of Lisbon to present the available courses, work developed inside and promote contact between university and high school students.

[3]Dia Aberto da Faculdade de Ciências, participação através do Departamento de Informática.
[4]Translation of the original title: "Classificação Interactiva de Filmes: MovieClouds e SoundsLike".

1.5 Planning and Schedule

The initial planning proposed for this project is presented bellow, with a total duration of nine months:

- Month 1 - 2: Gathering of requirements and characterization of state of art: Study of the subject's problems and related technologies or technologies currently being used, familiarity with previous work in the VIRUS project.
- Month 1 - 2: Writing of a preliminary report.
- Month 3 - 7: Proposal and development of interactive tools and mechanisms for movie access.
- Month 4 - 7: Development and testing of contents for the purpose of demonstration, evaluation and refinement of the elaborated work.
- Month 8: Evaluation and final refinements.
- Month 8- 9: Writing of the final report. Possibly writing of a scientific paper.

	Nome	Início	Fim	Duração
1	Inicio do PEI	17-09-2012 ...	17-09-2012 ...	0 dias
2	**Gathering of requirements and characterization of the state of art.**	**17-09-2012...**	**15-11-2012 ...**	44 dias
3	Study of the Problem	17-09-2012 ...	07-10-2012 ...	60 dias
4	Survey and research of related work	17-09-2012 ...	15-11-2012 ...	44 dias
5	Survey about technology currently under use	17-09-2012 ...	15-11-2012 ...	44 dias
6	Familiarization with previous work	17-09-2012 ...	15-11-2012 ...	44 dias
7	Writing of preliminary report	17-09-2012 ...	15-11-2012 ...	44 dias
8	Delivery of preliminary report	15-11-2012 ...	15-11-2012 ...	0 dias
9	Proposal and development of interactive tools and mechanisms for movie access.	16-11-2012 ...	18-04-2013 ...	110 dias
10	Development and testing of contents (for demonstration, evaluation and refinement)	16-11-2012 ...	19-03-2013 ...	88 dias
11	Evaluation and final refinements.	19-04-2013 ...	20-05-2013 ...	22 dias
12	Writing of the final thesis.	19-04-2013 ...	14-06-2013 ...	41 dias
13	Delivery of the final thesis.	15-06-2013 ...	17-06-2013 ...	0 dias
14	**Middleware Tecnologies (TM) Course**	**17-09-2012...**	**20-12-2012 ...**	69 dias?
15	Writing of 6 Reports (one per week) for the course	17-09-2012 ...	20-12-2012 ...	69 dias?
16	Course's project	01-11-2012 ...	20-12-2012 ...	36 dias?

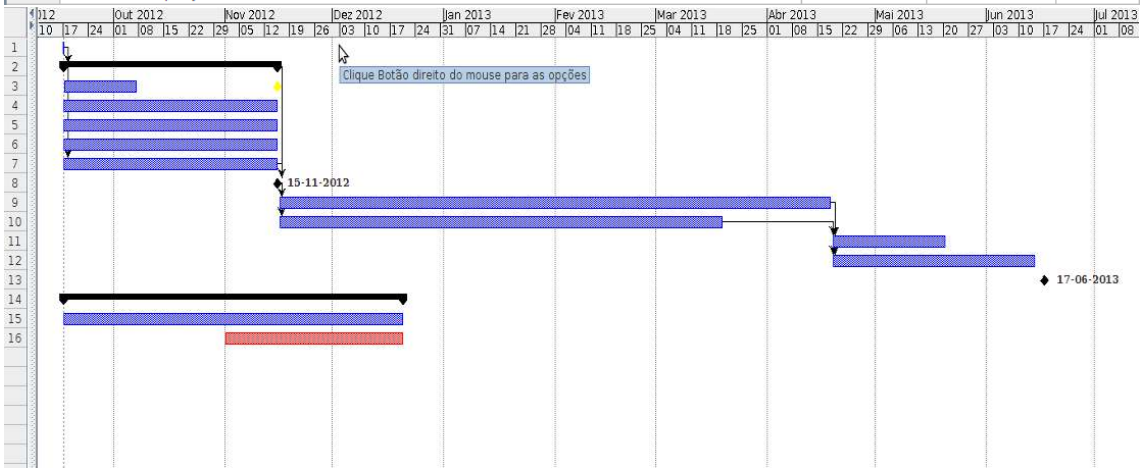


Figure 1.1: Schedule and Gantt map for the initial planning.

Due to the short duration of the project and the fairly well defined planning, the development process used in this project was the waterfall as a basis, with some iterations

and increments due to results of evaluation and the inherent research nature of the project. In this model, some phases are commonly defined such as *Requirements*, *Design*, *Implementation*, *Verification and Validation*, *Documentation and Maintenance*, where the *Requirements* phase corresponds to the first two months (until the delivery of the preliminary report), the *Design* and *Implementation* corresponds to the months 4-7, the validation to the month 8 and the *Documentation and Maintenance* to the last 2 months.

Figure 1.1 presents the original schedule and Gantt map for the project, involving the following tasks and milestones:

1. PEI start - A milestone marking the official start of the project;
2. Gathering of requirement and characterization of the state of art - the first part of the project, which is subdivided into the next four points;
3. Study of the Problem - Requirements gathering;
4. Survey and research of related work - Search and annotations about existing related work;
5. Survey about technology currently under use;
6. Familiarization with previous work;
7. Writing of preliminary report;
8. Delivery of preliminary report - milestone;
9. Proposal and development of interactive tools and mechanisms for movie access;
10. Evaluation and final refinements;
11. Writing of the final thesis;
12. Delivery of the final thesis - milestone;
13. Middleware Technologies (TM) Course - University course carried out in parallel;
14. Writing of 6 Reports (one per week) for the course - Reports for the TM course;
15. Course's project - A group project for the TM course;

It is worthy to mention that the work in the project, from September to December (four months) was carried out in parallel with the course of Middleware Technologies (*Tecnologias de Middleware*), which included lectures, weekly reports and a group project. This was already expected and accounted for in the initial planning.

I also participated in a workshop on Statistics Applied to Human Computer Interaction[5], that took place at DI/FCUL, on October 9 and 12 2012, with the further goal of applying the obtained knowledge in the analysis of data from usability tests.

Initially, the project was proposed with the usual duration of nine months. However the project took twelve months due to the time employed in the writing of more scientific articles than initially planned and the time required by the parallel course. Along time, the workflow was divided towards the tasks with more priority. In the case of scientific articles.

[5] "Workshop de estatística aplicada a HCI"

In figure 1.2, the final schedule is presented in a Gantt map for the following tasks and milestones. For a detailed and clearer map, please refer to appendix A:

Figure 1.2: Final schedule and Gantt map.

1. PEI start - A milestone marking the official start of the project;
2. Gathering of requirement and realization of state of art - the first part of the project, which is subdivided into the next four points;
3. Study of the Problem - Requirements gathering;
4. Survey and research of related work - Search and annotations about existing related work;
5. Familiarization with work already done;
6. Development of prototypes (audio similarity visualizations);
7. Writing of preliminary report;
8. Delivery of preliminary report - milestone;
9. SoundsLike - The second part of the project, which is subdivided into the next five points;
10. Proposal and development of interactive tools and mechanisms for movie access;
11. Development of SoundsLike Prototype;
12. Evaluation and final refinements;
13. Development of Video Visualizations;
14. Writing of Scientific Articles;

15. Writing of the final report;
16. Delivery of the final report - milestone;
17. Middleware Technologies (TM) Course - University course carried out in parallel;
18. Writing of 6 Reports (one per week) for the course - Reports for the TM course;
19. Course's project - A group project for the TM course;

1.6 Document Structure

This document is organized in the following order:

Chapter 1 - Introduction: The first chapter aims to be an introduction for the reader about this project, presenting the motivation, objectives and contextualization of the work;

Chapter 2 - Related Work: This chapter presents some of the related work public available in the academic world and on the web, and presents a summary of previous work done in the context of this project;

Chapter 3 - Movie Visualizations: In this chapter, we analyse the work done for the development of movie visualizations, including representation of movies along time and similarity between audio excerpts;

Chapter 4 - Movies Soundstrack Browsing and Labeling in SoundsLike: Here it is presented the main work towards the building of a prototype for the purpose of collecting data from movies soundtracks, including its purpose, requirements, design options and implementation details.

Chapter 5 - Assessing SoundsLike: This chapter presents a user study with the purpose of assessing and refining the prototype developed in chapter 4.

Chapter 6 - Conclusions and Future Work: This chapter summarizes the work presented across the entire dissertation, presents conclusions and directions for future work.

Chapter 2

Related Work

In this chapter, some studies, projects and ideas related to this dissertation are presented.

The first section presents some work and concepts related to MovieClouds (which is presented in section 2.6.2) surveyed during the first months on the project, that could be relevant in the design and implementation of SoundsLike (chapter 4) in future iterations. The other sections are more directly related to SoundsLike, by including references to and summaries of projects and studies related with the visualization of movies, information retrieval of audio and web applications for audio and video labelling.

2.1 Emotion and Representation Models

In psychology and philosophy, emotion is described as a generic term for subjective, conscious experience that is characterized primarily by psycho-physiological expressions, biological reactions and mental states [19]. It is linked to the arousal of the nervous system due to a reaction to external events, and cognition takes an important role in emotions, particularly in the interpretation of such events [15, 19].

A certain music or film can induce the experience of a diverse range of emotions in the audience. A sudden moment of danger or tension can trigger an involuntary muscular motion in most of the observers sitting in a cinema, due to a natural response of the human brain to a possible threat, and may cause the experience of fear. A sad moment in a film could trigger emotions of sadness or despair in the audience, but depending on the life experiences and personality of each individual, not everyone reacts in the same way - some may cry, others may not - since an interpretation of the event is taking place inside the brain that accounts for all the past experiences.

Emotions are hard to describe, not only due to their abstract and complex nature but also due to the ambiguity introduced by language itself. Plutchik refers to people not knowing the difference between emotions, such as fear an anxiety, guilt and shame or envy and jealousy, and often using metaphors for describing such perceived emotions.

Scientists proposed more than 90 definitions of "emotion" during the 20th century [52],

but there are two aspects about which most of them agree in psychology studies: emotions are the result of reactions to relevant events for each individual's needs or objectives, and emotions are related with psychological, emotional, behavioural and cognitive components [5]. A direct comparison between humans and animals is made in Darwin's Theory of Evolution, where he states that expressive emotional behaviours are used to transmit information from an animal to another and therefore play an important role in the survival of each individual [25].

A quick summary of relevant emotional models and studies about the relation between colours and emotions is presented in the next sub-sections.

2.1.1 Circumplex Model

For years, scientists formulated diverse ways of organizing the different emotion names available in logic representations. Some attempted to create simple representations that would organize these emotions under dimensions of pleasure-displeasure, arousal-sleep (degree of arousal), dominance-submissiveness (related to potency).

Russell formulated a model using a Cognitive Structure of Affect where eight emotional variables fall into a two dimensional space. Four of these variables are related to the axis: the horizontal axis represents the pleasure-displeasure dimension (valence) and the vertical one relates to the arousal-sleep dimension (arousal). The remaining four variables do not form independent dimensions but are used to name the quadrants of space where an affect is defined by combinations of the axis dimensions [55].

In figure 2.1, the left diagram shows these eight affect concepts in circular order, the right diagram displays a direct circular scaling coordinates for 28 affect words.

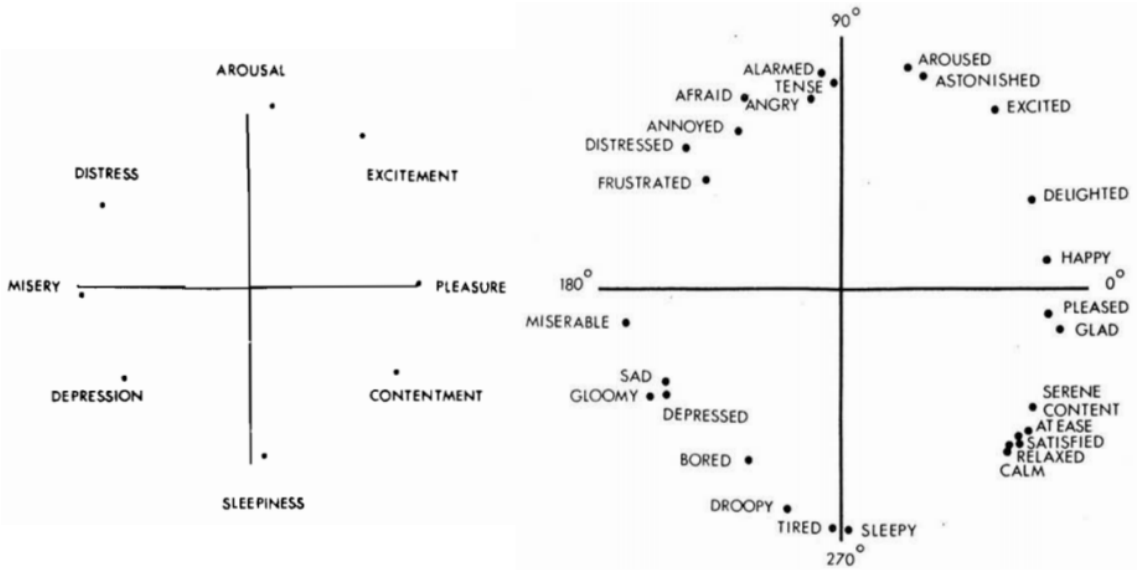


Figure 2.1: Russell's Circumplex Model of Affect or Emotions [55].

2.1.2 Ekman Basic Emotions

Paul Ekman is an American psychologist who has been a pioneer in the study of emotions and their relation to facial expressions. In [13], he proposed two appraisal mechanisms for stimulus and emotional responses: An "automatic" mechanism that reacts quickly and selectively to external or internal stimulus resulting in a emotional response, sometimes without the subject's awareness. Ekman postulates that this mechanism must act automatically (involuntarily). A "extended" mechanism that requires cognition, where an evaluation of the environment and surroundings is needed, where such evaluation can be slow, deliberated and conscious.

From the results obtained by his studies of facial expressions, Ekman proposed 6 emotions as basic: "Anger", "disgust", "fear", "happiness", "sadness" and "surprise". Basic emotions are claimed to be biologically fixed, innate and as a result universal to all humans and many animals as well [14].

Later, Paul Ekman expanded the idea of basic emotions to a "basic emotions framework" composed by 16 emotions that shares characteristics that "distinguish basic emotions one from another and other affective phenomena" [15]. These emotions are "amusement", "anger", "contempt", "contentment", "disgust", "embarrassment", "excitement", "fear", "guilt", "pride in achievement", "relief", "sadness/distress", "satisfaction", "sensory", "pleasure" and "shame", where each word can be used to represent a family of emotions allowing the expansion of the list. Ekman denotes about omitting "some affective phenomena which others have considered to be emotions".

2.1.3 Plutchik's Model of Emotions

Plutchik analysed hundreds of emotion words available in the English language and concluded that they fall in diverse families based on similarity to primary emotions. These primary emotions were organized in a colour wheel (circumplex model), where similar emotions were placed close together and opposites 180°apart. The remaining emotions have been considered a mixture of the primary emotions.

He reused the notion of a circumplex model from previous work of Willian McDougall (1921) and Harold Scholosberg (1941), and expanded it like a structured model, by adding a third dimension related to the intensity of the expressed emotions, in a cone format shape, as can be seen in Figure 2.2.

The following primary emotions were chosen by Plutchik as basic biological emotions for his model: surprise, fear, trust, joy, anticipation, anger, disgust and sadness.

In Plutchik's model, it is also proposed a relationship between emotions and colours by association of brighter and warm colors with positive emotions such as happiness and excitement (yellow), and dark and cold colours with negative emotions such as sadness , boredom and anxiety (blue) [52].

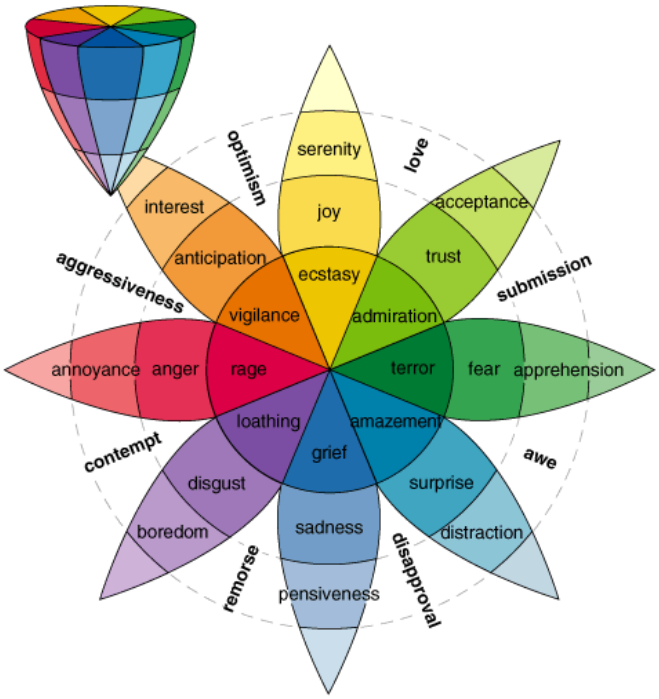


Figure 2.2: Plutchik emotional model.

2.1.4 Colours and Emotions

"Colours, like features, follow the changes of emotions." - *Pablo Picasso*

Parallel to psychology and philosophic studies of emotions, the relation between colours and emotions have been a subject of interest between investigators and artists. Always interested in improving their work, artists look out for new ways of expressing and inducing emotions in the viewers, and the colour combination has a strong influence in all artistic work and the expressed message.

Colours can be used in diverse contexts: a paint of a cosy place may use warm and bright colours to express safety and comfort; a sad set in a drama or music show would include lighting with pale and cold colours to express melancholy; prohibition and danger signs use bright red tones for grabbing a person attention; etc.

believe in the effect of colours on human mood and emotions. There are diverse formulations and theories about the impact of colours on people and their emotional states. In most of these studies there is a common agreement about the meaning and impact of each colour.

Each colour is not only associated with one specific emotion, it is able to induce distinct emotions and moods depending on the context. Most of the time these triggered emotions can be similar or related, but sometimes they are totally opposite. For instance, red is associated with energy, desire and passion, three similar and related emotions, however it is also associated with love an hate, interestingly two totally opposite emotions.

"There's a thin line between love and hate
Wider divide that you can see between good and bad
There's a grey place between black and white
But everyone does have the right to choose the path that he takes"

- Iron Maiden

Lets quickly enumerate meanings and emotions evoked by some colours:

- **Red**: Colour of the human blood and fire, provides a sensation of energy and warmth. People often associate red with danger, strength, power, determination, desire, passion, love and hate. A very emotionally intense colour used to stimulate the body and mind, and to increase circulation. A colour used to grab the person's attention in danger signs and traffic lights, in the "click here" or "buy now" buttons and advertises of websites, in advertisement to evoke erotic feelings or to promote energy drinks, games, cars and items related to sports.
- **Pink and Purple**: A colour associated with tranquillity, femininity, sweetness, warmth and love. Light pink evokes positive, romantic and nostalgic feelings while dark purple evokes gloom, sad or frustrating feelings.
- **Yellow**: The colour of sunshine, and associated with joy, happiness, intellect and friendship. Causes a warming effect and is associated with the stimulation of mental activity and purification of the body. Used to highlight objects, attract attention in warning signs and to symbolize honour and loyalty in heraldry, but it can have a disturbing effect when overly used and can sometimes be connoted to cowardice.
- **Green**: Colour of nature and associated with growth, prosperity, fertility, health, tranquillity and jealousy. It is considered one of the most restful colours to the human eye and has a calming effect in the mind. Used frequently in the advertisement of drugs and medical products, and commonly associated with ambition, greed and money. Can also be used to denote the lack of experience, and together with yellow can mean sickness, discomfort and cowardice. In heraldry, green indicates growth and hope.
- **Blue**: Colour of the sky and sea, associated with depth, stability and with the intellect, but provides a cooling effect that relates it with tranquillity, melancholy, grief, sadness and sometimes anxiety (depression). Commonly used in products and services related with liquids, cleaning, air and sky (water, drinks, cleaning products, voyages), to suggest precision in promotion of high-tech products, and it is considered a masculine colour.
- **Black**: Colour of the night and the absence of colour, associated with power, elegance, seriousness, death, mystery and fear. It denotes strength, elegance and authority, but usually has a negative connotation being related to the unknown and the unexpected, also correlated to fear, loss, mourning, grief and death. Used to make

others colours stand out and to create a contrast with brighter colours. Symbolizes grief in heraldry.

- **White**: Considered the colour of light and snow, associated with goodness, innocence, purity, virginity and perfection. This colour stimulates a sensation of cleanliness, simplicity and safety and is commonly used in high-technology products, low weight/fat food and in medicine, including medical products advertisement to suggest safety.

2.1.5 Emotionally Vague

Emotionally Vague[1] is a study about emotions and how people react and feel when they experience them. The study also focus on the relation between the body and the emotion, including descriptions of feelings using words and colours. The method involved the formulation of 5 objective questions, related to 5 pre-chosen emotions (Anger, Joy, Fear, Sadness and Love): one for describing the reason for feeling each emotion using words, three questions about the location of the feeling in the body, the exact spot and the direction of it, and another question about the colours that we associate to the emotions (the subjects had to use a fixed colour palette for referencing colours). A sample of 250 individuals has used [49].

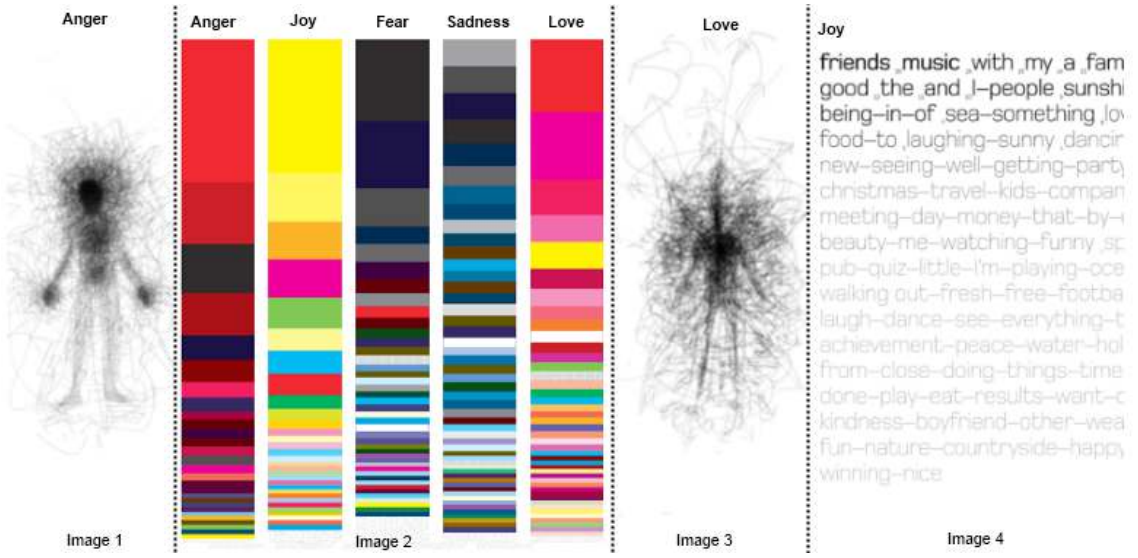


Figure 2.3: Emotionally Vague: 1) Overlaid of the answers about how individuals feel the Anger emotion. 2) Most chosen colours associated to each emotion. 3) Overlaid of the answers about the perceived direction for Love emotion in individuals' bodies. 4) The highest frequency of words extracted from the answers where individuals described feelings generated by the joy emotion.

One of the interesting results about this study are those about the colours and their relation with emotions.

[1] http://www.emotionallyvague.com

In Figure 2.3, it is observable a specific pattern of colours for each emotion: In the first image, the subjects had to draw everything they wanted about here and what they feel when experiencing the emotion Anger, from their memory, resulting in a image where people perceive experiencing the emotion with more emphasis on the head, followed by the chest and hands. The second image represents association of colours, obtained from all subjects, related to the five emotions Anger, Joy, Fear, Sadness and Love. In the third image, subjects were asked to draw on paper the direction of felt emotions in their bodies, being this image the representation of all results for Love emotion, where most individuals represented as leaving the chest area, and directed the outside. The last image shows the combination of all words that subjects associated to the Joy emotion.

Anger was mostly associated with tones of red and other dark colour tones, joy with yellow and miscellaneous softer tones, fear was related to black and gray tones, sadness had one of the most associated number of colours (no specific dominant colour) and relates to multiple gray and blue colour tones, and love was associated with red, pink tones and other soft colour tones. It is observable that Anger and Love relate mostly to warm colours, while Fear and Sadness relate to neutral and cold colour tones, and Joy also relates mostly to warm colours and to some soft and bright cold colours (blues and greens). Notice how red is the dominant colour in two opposite emotions - love and hate. This study may be interesting to help developers to choose colours for interface elements related to specific emotion, e.g. relate an action with a specific emotion to alert the user (irreversible delete action represented by a red icon).

2.1.6 ColorsInMotion

ColorsInMotion is an application that supports interactive and creative visualization strongly based on colour and motion properties, offering new ways to visualize and explore video in 2D. The main introduced innovation is enabling users to visualize the video space through different views, allowing them to search, compare and interact with a selection of videos in a cultural context, featuring areas such as music and dance from different authors and countries.

It features six main views for visualization. In the movie spaces, videos are presented as a physical particle system where a collection of video icons move and group together on the screen according to their colour similarity and user interaction. These coloured clusters display zones from the video space composed by sets of video with similar colours, and distant from videos with different colours.

In figure 2.4 the available views are observable: the first view (a) displays the video loops which includes frames taken at constant time intervals, working as content overview or summary. The second view (a) represents videos through coloured circles composed by the dominant or average colour from each one. The third view (c) represents movies not only by their colours, but through a set of rectangles with coloured stripes featuring

the proportion in which they dominate inside the video.

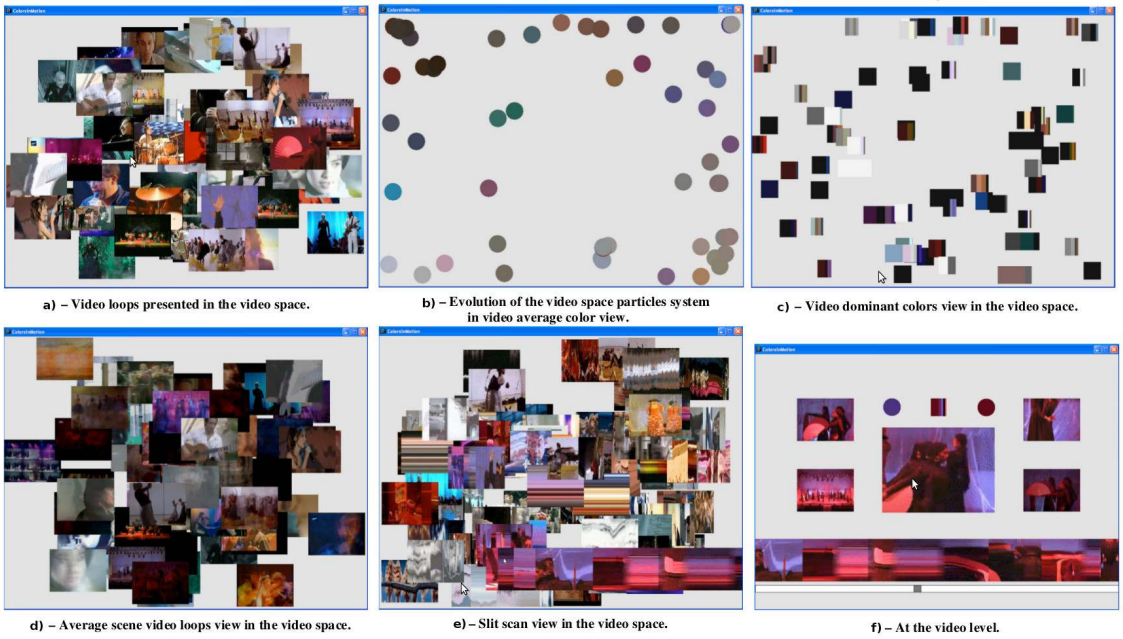


Figure 2.4: Different Views in ColorsInMotions.

The fourth view (d) emphasis the motion aspects of video by representing them with image loops, where each image captures the movement through the average of pixel colours occurring in each scene from the video.

The last view (e) displays static slit scan images that represents the movement from the movie that have been taken off. A slit scan is a technique to capture the action happening in the video centre, along time, and represent the result sequentially on a static image. This generates a movement effect visible in figure 2.4:e.

Any view allows direct access to any video present in the movie space. By accessing a video, we leave the global space to enter in the video level. At the video level (2.4:f), it is possible to view the entire video, the details and different views. Here, the video is playing the centre. In the top row, the colour aspects are highlighted through the average colour loop and circle (top-row-left), followed by the dominant colours rectangle and the dominant colour circle (top-row-right). To the left of the video, the traditional loop, to the right we have the average scene loop stressing movements aspects and complemented with a scrollable slit scan in the bottom.

It is also possible to search and select videos based on cultural contexts, through average or dominant colour. This search can be achieved by selecting one or more colours from a palette or from the real world through a web-cam. The search can be issued from any of the video spaces views. Results are presented in the same view where the search was issued.

This system provides new ways to visualize and browse movie spaces but do not

includes emotions, sound events and other video dimensions. Searches and navigation are constrained to the colours and movement properties extracted from videos.

2.2 Visualization And Classification of Audio and Video Information

"Visualization is defined as the use of computer supported, interactive visual representations of data to amplify cognition, helping people to understand and analyse data", turning "empirical observations into explanations and evidence" [7, 59].

There are different ways for representing data to the user. In [32], some important concepts guidelines are presented that should be followed during the design and creation of data visualizations. These include the importance of balancing the quantity of displayed information vs complexity, the use of overviews for giving the user a sense of context and help finding information, the possibly of choosing and getting more details about chosen subsets of information (e.g. zoom) and different levels of information in order to provide higher detail and maintain coherence, the importance of data comparison by displaying relationships between items, the use colour techniques to address multidimensionality and the importance of sorting information by time and space allowing the representation of movement and info flow, besides being considered hard to handle. It also presented a structured criteria for dealing with representation of temporal dimension.

This section presents some concepts, projects, studies and applications about visualization and classification of audio and video.

2.2.1 Twitter Lyrics

Twitter Lyrics [3] is a project aimed for alternative visualization and popularity analysis of songs based on user preferences instead of resorting to the music industry rating system.

It uses Twitter, a worldwide social network built around the concept of short status messages nicknamed as Tweets, as a source for obtaining data about the popularity of music worldwide. Twitter Lyrics expects to gather a dataset of trends and sales for future comparison with the ratings provided by the top music industry charts. These trends can be obtained by exploring the number of times a song lyrics were quoted in status messages, obtaining an "accurate insight into the perceived value of the song".

Figure 2.5 displays two of the public images disclosed of the project's analytics tool, showing some trend graphs and visualizations obtained from Twitter data.

What makes the project even more interesting is the capability of perceiving the real emotion that these musics evoke and associations created in the people's minds by analysing comments posted in a real social network, by users, quoting lyrics from songs, features not present in music industry ratings and charts, where they currently use the al-

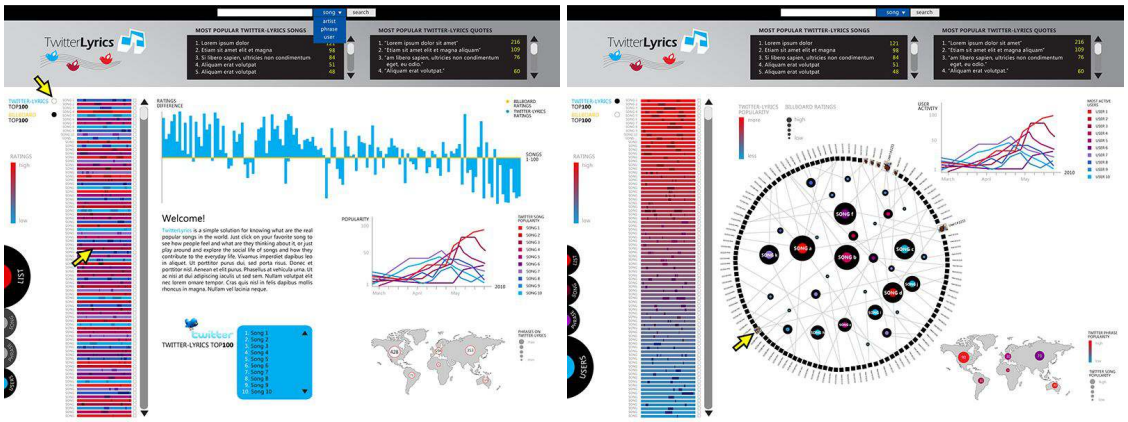


Figure 2.5: Twitter Lyrics Demo Version Screenshots

bums sales and commercial success of artists as the source of information. Twitter Lyrics information could be used to facilitate the bounding of people through these associated and evoked feelings and might provide an alternative perspective of the music industry [3].

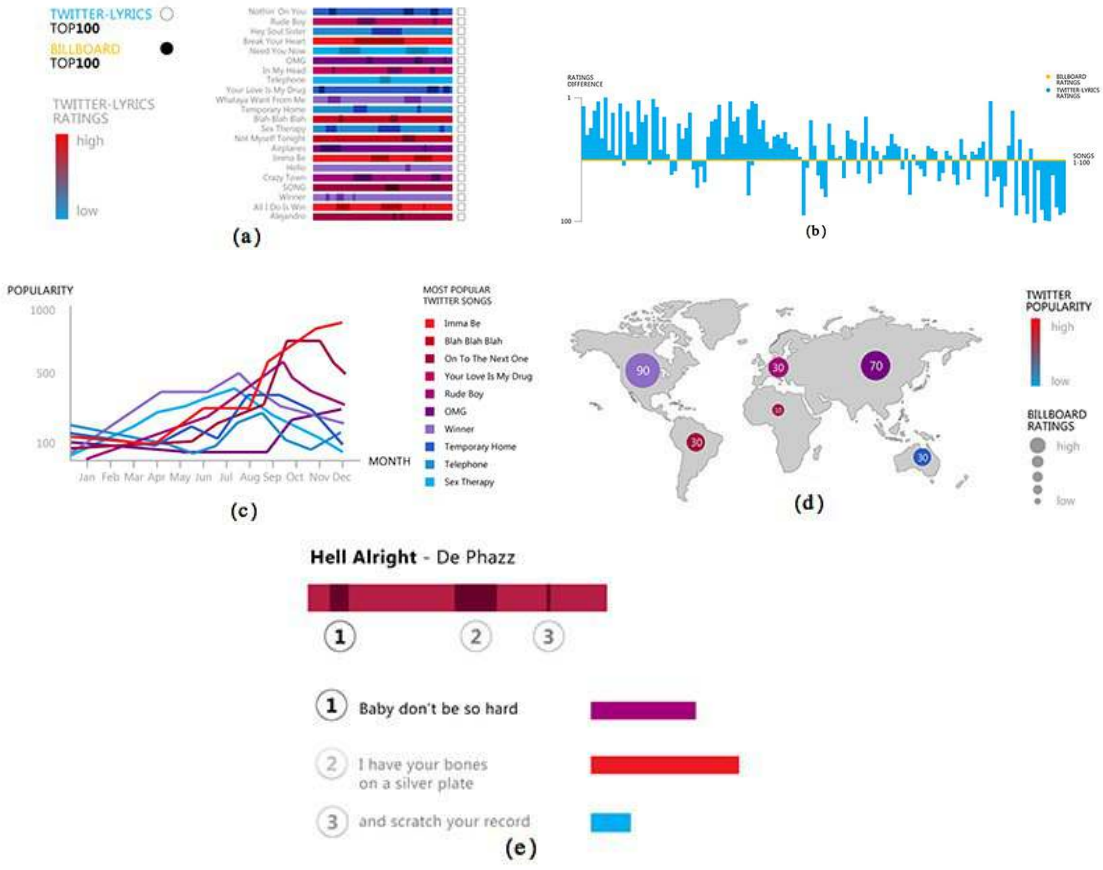


Figure 2.6: Twitter Lyrics Visualizations

In figure 2.6 it is possible to observe different kinds of visualizations, including bar graphs to display differences between selected charts (b) or about a song or a phrase (a),

line charts for observing changes in popularity of a set of songs or phrases (c), trend graphs displaying the geo-localization of users (d), and a bar visualization with timeline for displaying popularity of phrases within a song (e).

2.2.2 Last.fm

Last.fm[2] is a music recommendation service where users can receive personalized recommendations based on their current listening habits. Being also a social network, Last.fm users can also comment on other users profile (shouts), mark the songs that they love the most or ban them from their library, comment on an artists/album/track page and comment on user groups, and add textual tags to the artists, albums and tracks.

2.2.2.1 Recommendations and Scrobbling

Some of the features previously mentioned are publicly accessible. However, without a user account the only possible recommendations are based on artist similarity. An account is required to submit new comments and tags, mark songs, and to use the recommendation system.

The recommendation system requires user information to provide good recommendations. A user can use the system by adding artists to the profile library and obtain recommendations based on artist similarity and artists tags.

To improve the results obtained from the recommendation service, the user should configure a utility program available on the site[3], titled *The Scrobbler*, which reports to Last.fm every music listened, providing the listened track name, album and artist. Each entry is denominated as *Scrobble*.

Scrobbling[4] enables the tracking of the users listening habits and provides a unique and highly personalized recommendation results for every user. Users can also obtain statistical data from their habits and easily compare them with other users.

2.2.2.2 Last.fm Tagging System

In Last.fm, users can label artists, albums and tracks with textual tags. In each artist/album/track page, the most common tags are displayed and sorted by number of users. Tags are usually used to describe the music by its genre, but you can find tags about expressed emotions, music tempo or melody, lyrics idiom, artist group country, and other miscellaneous subjects.

Tags act as a categorical classification for the whole artist/album/artist, reflecting the interpretation of users about the listened music. Tags also take an important role in

[2] `http://www.last.fm/`

[3] As alternative to the official Scrobbler application, many music players and platforms provides integration with Last.fm natively or through plug-ins.

[4] The act of sending Scrobbles to Last.fm.

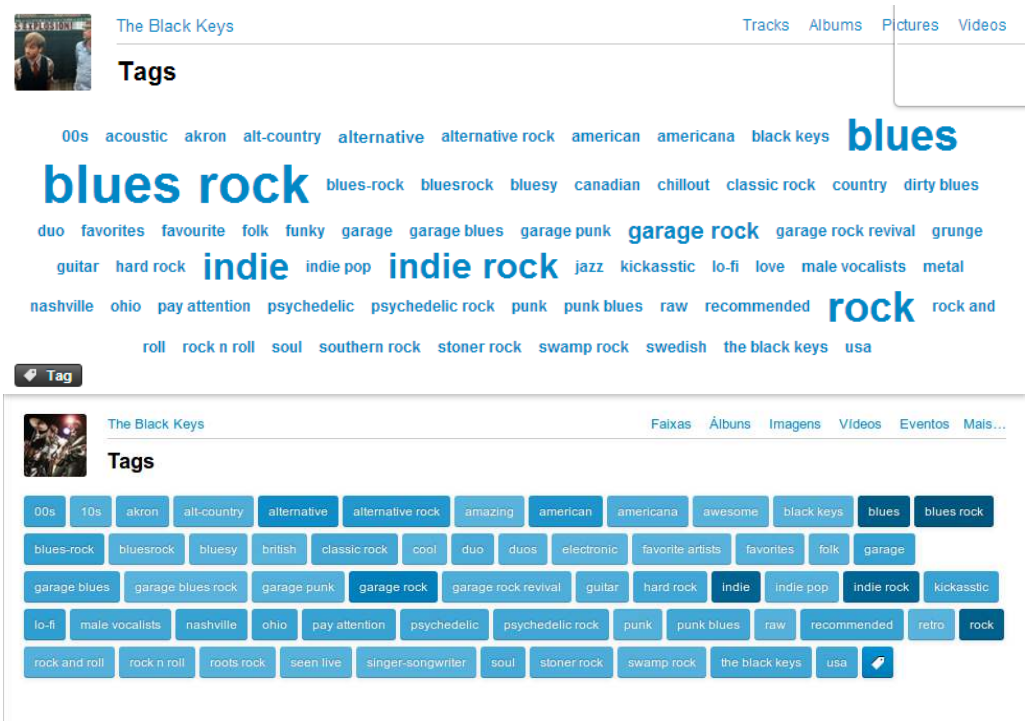


Figure 2.7: Last.fm tag cloud for the artist "The Black Keys". On the top, the old last.fm design using a classic weighted tag cloud paradigm, on the bottom the most recent design using the same paradigm but displaying weight through colour intensity.

the last.fm recommendation system and other Last.fm services [36]. In 2007, Last.fm database contained more than 1 million unique free-text labels used to annotate millions of songs [61].

With the rising community of Last.fm users, the number of tags is also increasing at an exponential rate, allowing the extraction of information about instrumental characteristics and common emotional reactions from these tags. Last.fm provides a REST webservice API to access all stored information, including tags.

All tags are assigned freely by the users with valid user account. To create an account, users must provide a username, password and an existing email address (which is verified later). There are no specific guidelines for music or artist classification and not everyone is a musical expert, thus, it is expectable to have users incorrectly classifying artists because their musical expertise is not high enough or due to others factors like the user's current motivational state or disposition.

Users can also provide incorrect tags on purpose, a single user cannot have a negative effect in an artist classification, because their tags have no practical weight or importance by counting only as one vote, becoming outliers, unless the music or artist do not have a significant number of tags.

With a significant number of tags given by dozens, hundreds and even more different users, common tags will stand out of the remaining tags, and those common tags can

be used as information for music classification, since they are the product of common agreement between the Last.fm users community.

2.2.2.3 Last.fm Discover

Last.fm Discover[5] uses the recommendation system and tagging systems to allow the users to explore new music using a tag wave visualisation (Figure 2.5). The main view shows a landscape composed by four layers of small green hills and a sunrise in the horizon. Each layer of hills displays a group of music tags from the database, whereas the nearest layer displays the most significant tags and the farthest layer displays related tags to the ones in the previous layer.

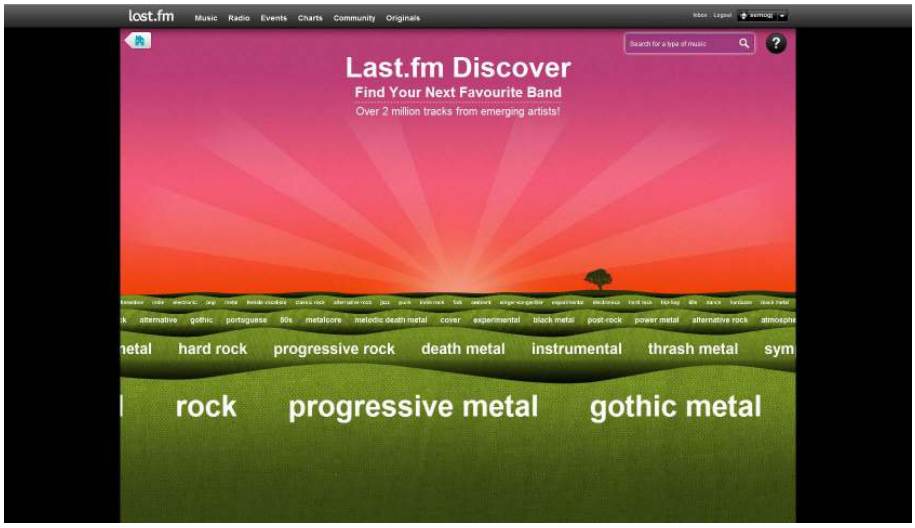


Figure 2.8: Last.fm Discover Main View

The initial shown tags are chosen based on the user's recent listening habits and the user must pick a tag. When the user selects a tag, the layer's content refreshes and the new tags are related to the current selected tag. The user is now allowed to play music tracks directly related to the selected tag.

When the user clicks "Play", the layers refresh again and instead of tags, they display music tracks names, album image and artist in each layer, and a music player for playback of the music in the first layer (Figure 2.9). The user has the option to skip ahead into another similar artist's track, select a track in one of the layers, update the layers with tracks similar to the current playing song, or skip and update the tags with different music style within the initial selected tag.

[5] http://www.last.fm/discover/

Figure 2.9: Last.fm Discover Album View

2.2.2.4 Mood Report For Last.fm

With the huge musical dataset obtained from the users, the Last.fm team has been actively working on many projects to extract useful information from the artists information, artists and music tags, and users listening habits to improve the recommendation system and the radio service.

One of these emerging and promising projects is the Mood Report[6], a stream graph based visualization that displays the music mood from the listening history (submitted Scrobbles) of a Last.fm user. Examples of mood tags include 'Aggressive', 'High Energy', 'Punchy Dynamics', 'Fast', 'Slow', 'Strong Dance beat, 'Smooth', 'Happy', 'Sad', 'Relaxing', 'Ambient', etc.

It is not clearly stated how and why these specific tags where chosen but it is closely related to a study from the same author in [37], where he concludes that social "tags define a low-dimensional semantic space that is extremely well-behaved at the track level in particular being highly organised by artist and musical genre", and introduces Correspondence Analysis for the visualization of such semantic space and how it can be applied to display musical emotion. In the same article, the author refers to some emotional representation studies. The tags weights are obtained automatically from extracted information from a music analysis system, built by Last.fm, which uses user feedback from three different demonstrations[7] [8] [9] for confirmation of the results and optimization of the automatic analysis [36]. This visualization, observable in Figure 2.10, lets users observe mood changes in their musical life over the past 120 days. The graph height represent all

[6] http://playground.last.fm/demo/moody
[7] http://playground.last.fm/demo/speedo/
[8] http://playground.last.fm/demo/complexity/
[9] http://playground.last.fm/demo/evaluator/ate/

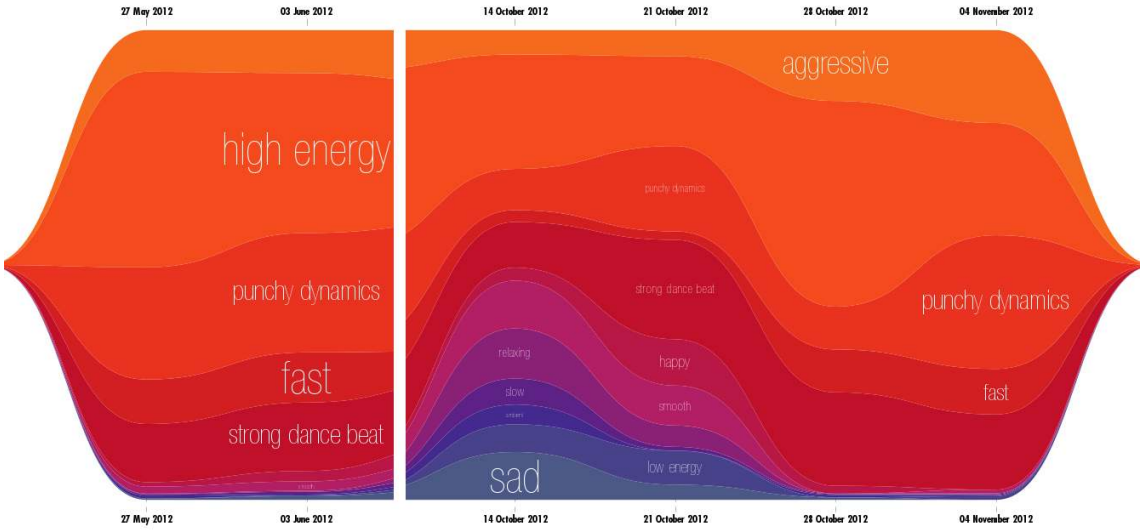


Figure 2.10: An example of a Mood Report from an active Last.fm user

the user's listened music normalized to a fixed height grouped by weeks.

Besides being far from perfectly accurate, due to the difficulties of automatic tagging [36], in the future, users will have a potential benefit from a highly adaptive music recommendation, radio streams and automatic music playlist systems. They will adapt not only to their current musical taste, but will take into account the expressed mood in all audio tracks, improving the user listening experience by making possible a matching of the music mood and other characteristics with the current user mood. For instance, users in a bad mood or experiencing tense or depressing moments will be able to request suggestions of happy and calm music tracks to influence their current mood in a positive way (e.g. a writer working on a sad novel may try to find and listen to sad and depressing music to get in the correct mood to obtain inspiration). Such system could be implemented in any audio player, being only necessary to have a local or remote mood database where the track mood could be obtained, and Last.fm is a candidate for this job.

2.2.3 The Structure Of The World-wide Music with Last.fm

A example of what can be done with Last.fm data is the representation of all artists in a map highlighted their similarity relationship as it can be shown in figure 2.11 [47].

The project's author crawled the Last.fm database using the Audioscrobbler API and obtained a list of artists, artist relationships and tags found. The crawl was conducted by starting from an arbitrary artist (in this case, Nightwish[10]) and by expanding the search to its similar artists. Some measures have been taken to prevent erroneous data, for example including artists without a MusicBrainz[11] ID. But the author notes these measures also

[10] `http://last.fm/music/Nightwish`

[11] MusicBrainz is an open music encyclopedia that collects music metadata and makes it available to the public. `http://musicbrainz.org/`

could prevent the inclusion of legitimate musicians who are not popular enough or do not possess a MusicBrainz ID, and artists who are not reachable through similarity path from the starting artist.

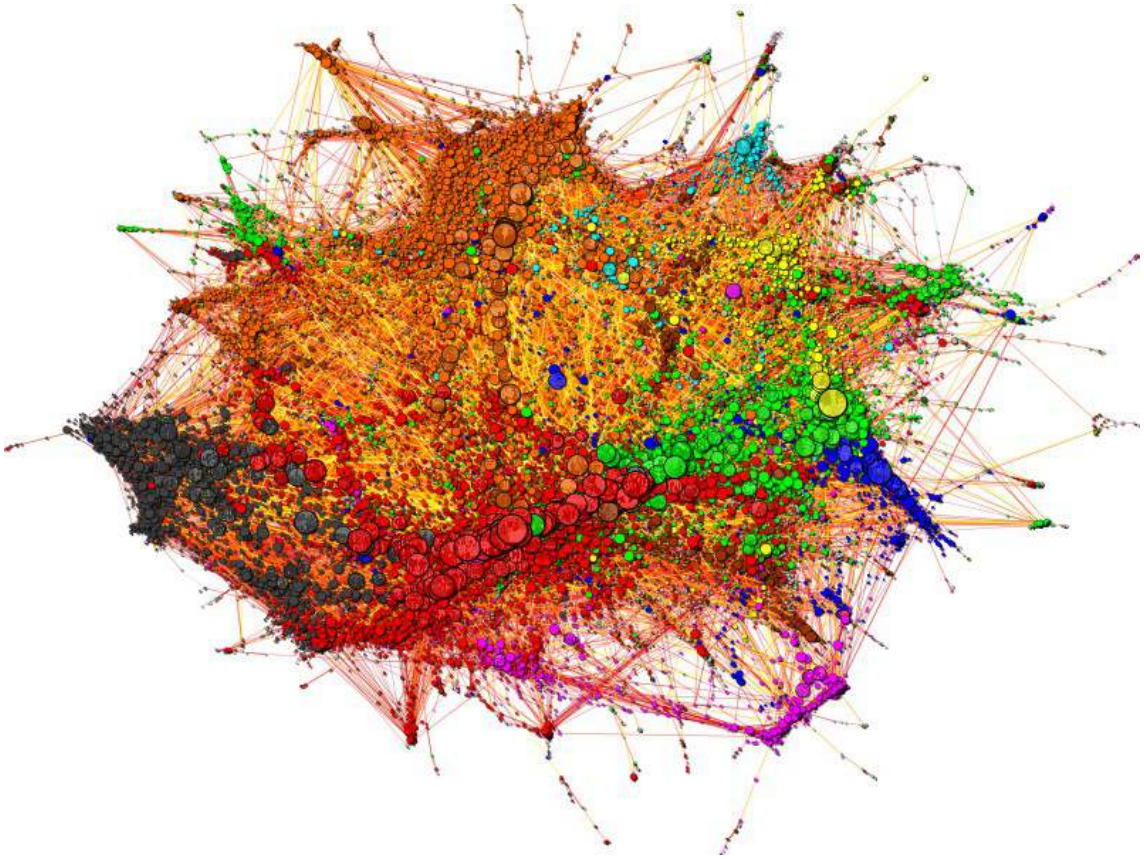


Figure 2.11: Artist's Map Graph of Last.fm

Each vertex (node) in the graph represents an artists and each edge (line) represents a similarity relation between two artists. The nodes size are proportional to the number of listeners of the top track of that specific artist. The nodes colours represent information about the artist main musical genre inferred from the top tags (most common tags) that have been attributed to the artist: Red is related to generic rock music types including classic, punk, hard and alternative rock; Green is related to pop, soul and funk; Black is related to heavy metal music and all sub-genres; Orange is related to electronic music and all sub-genres; Dark Blue is related to hip-hop and rap music, Yellow relates to jazz, brown relates to country, folk and world music, Light Blue is related to classical music; and Pink relates to reggae and ska music.

The edges colours are chosen according to their betweenness centrality[12] scores in the entire node network. [58]

[12]Betweenness centrality is a measure of a node's centrality in a network. It is equal to the number of shortest paths from all vertices to all others that pass through that node.

2.2.4 Audio Flowers: Visualising The Sound Of Music

What makes an arbitrary music track so different from another even if composed from the same musical instruments? In Music Theory, there are diverse characteristics, elements, parameters or facets that composers can manipulate to create a huge set of unique music pieces, and that most of them can be distinguished by the common listener (Boretz, 1995). The fundamental elements of music include pitch, rhythm, melody, harmony, texture, timbre, structure, tempo, etc.

Changes to each of these elements will be noticed by the listener. For instance, different versions of the same song can have differences in certain musical elements such a richer rhythm or pitch [44]. Pitch, which is based primarily on the frequency of vibration, changes depending on the source of the sound, thus each musical instrument has a particular pitch, and such difference allows for the listener to identify the source of a specific sound even in the middle of many others (e.g. a bass guitar sound in a music track).

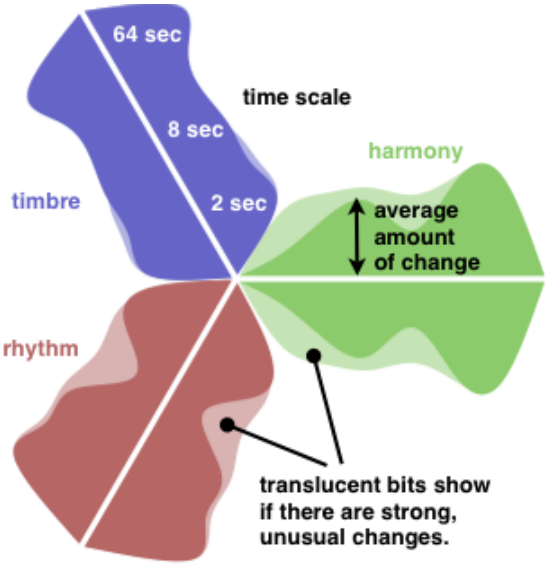
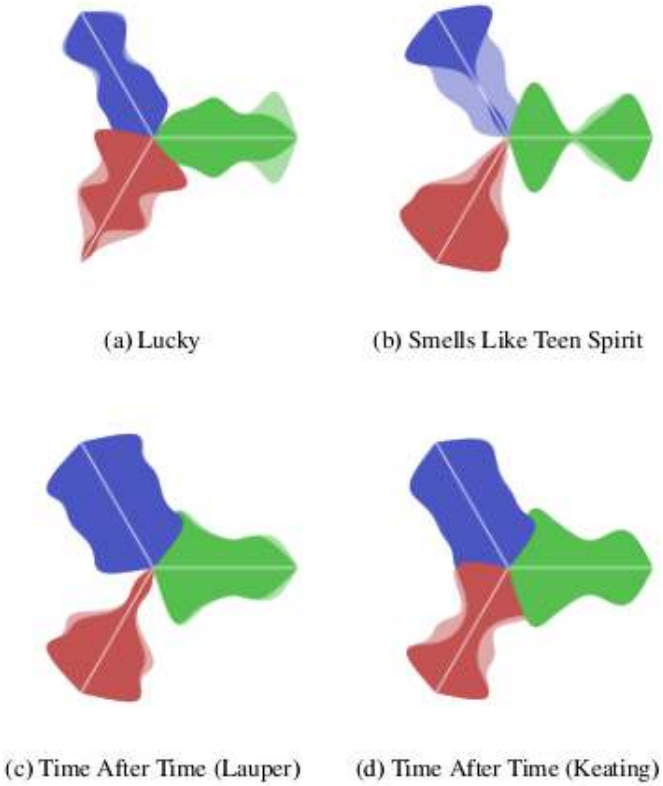


Figure 2.12: Audio Flower Representation

Another project from Last.fm team is a compact visualization of music structural changes of Rhythm, Harmony and Timbre in a flower like graph: Audio Flowers [44].

The visualization is composed by 3 petals. Each petal represents a fundamental element: red for rhythm, blue for timbre and green for harmony. and created from two overlaid graphs (an opaque and a translucent graph) obtained from a plot of the calculation of a 100-point smoothed interpolation of values obtained from an audio source (e.g. a music file) analysis, where the closest value to the centre of the representation is the start of the sound track (short time scale) and the tip of the petal the end of the track. The wider the petal is the greater the difference in the specific musical element represented by its colour, being the opaque graph the normalised median values and the translucent graph the normalised mean values. Figure 2.12 illustrates the proposed Audio Flower representation [13], exemplified in Figure 2.13 with four different music tracks [44].

[13] Image obtained from `http://playground.last.fm/demo/complexity`



(a) 'Lucky' performed by Britney Spears
(b) 'Smells Like Teen Spirit' by Nirvana
(c) 'Time After Time', by Cyndi Lauper
(d) 'Time After Time', by Ronan Keating

Figure 2.13: Audio Flower Samples

2.2.5 Netflix Prize Similarity Visualization

"Netflix[14], Inc. is an American provider of on-demand Internet streaming media in the United States, Canada, Latin America, the Caribbean, United Kingdom, Ireland, Sweden. It started its subscription-based digital distribution service in 1999, and by June 2013 offers more than one billion hours of TV shows and movies per month to more than 73 million of users across 41 countries [28, 53].

The Netflix Prize was an open competition held by Netflix aimed for finding the best collaborative filtering algorithm for predicting user ratings for films based only on previous ratings. These ratings were just quadruplets in the form of <user, movie, date of grade, grade>, being the grade values between 1 and 5 (stars). Netflix provided huge datasets of information: A Training, Probe and Qualifying sets, each one was composed by different ratings to make difficult (and prevent) hill climbing on the test set.

The real challenge of Netflix Prize was the creation of a recommendation system from a dataset with characteristics of holding too much data and too little data. The datasets are massive, making available too much data to be handled by simple analysis techniques

[14] https://www.netflix.com/

or to simply browse it, but there is too little data for making accurate recommendations for all films - the data is far from being normal and there is a huge quantity of users that have rated few films and films with a lower quantity of ratings, making predictions hard to make for these cases.

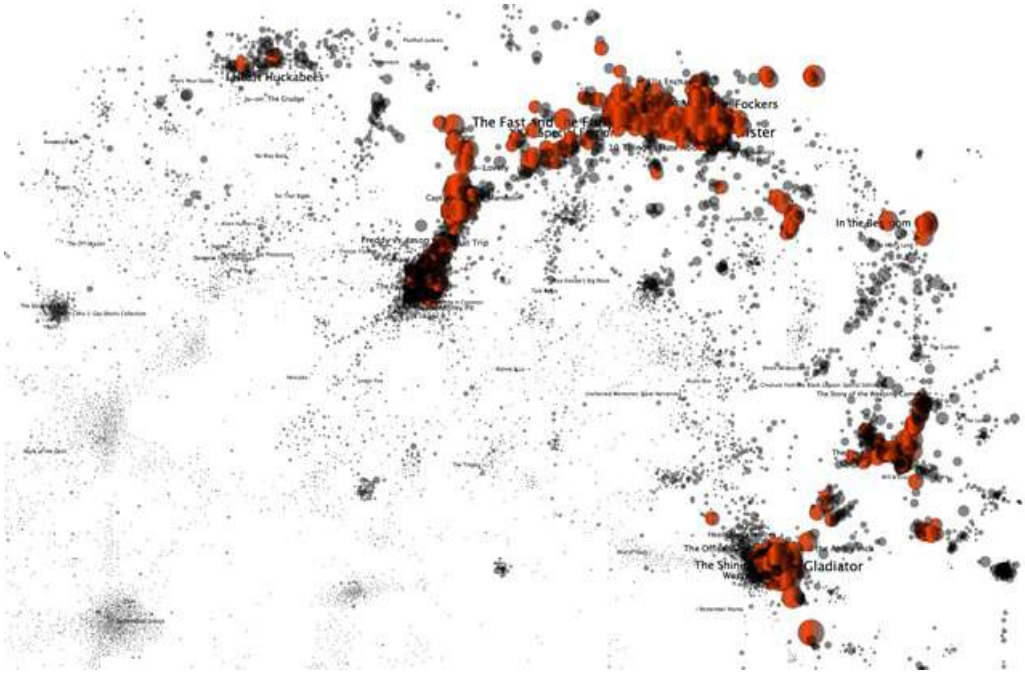


Figure 2.14: Visualization of the 17.000 movies in the Netflix Prize dataset

Some visualizations were proposed using similarity clustering algorithms and a diverse range of labelling strategies. The visualization in figure 2.14, obtained from the Netflix datasets, shows clusters of films grouped by similarity in a vast disperse cloud of films. The huge displayed dispersion is due to the fact that the test data was not statistically uniform (far from being represented through a normal distribution). It displays a dispersion of movies with some cluster of movies, where its observable that these clouds represents similar movies and with a great probability to belong to the same genre.

2.3 Content-based Audio Retrieval With Relevance Feedback

Relevance Feedback algorithms are used in information retrieval systems, for using user's judgements on previously retrieved information to construct a customized query for future requests. "These algorithms utilise the distribution of terms over relevant and irrelevant documents to re-estimate the query term weights, resulting in an improved user query" [26]. It has been studied in a diverse range of applications and has proven to be an effective method to increase performance in diverse fields of information retrieval, such as video [69], image [30] and text retrieval [8].

In the last 10 years, audio retrieval began as a relative new research branch, but has been lacking user interaction. More recently we have witnessed the increasing of musical streaming sites using feedback features such as Jango[15], Grooveshark[16] and others, in their streaming radio and recommendation systems, where musics are suggested based on the users' taste and previous feedback.

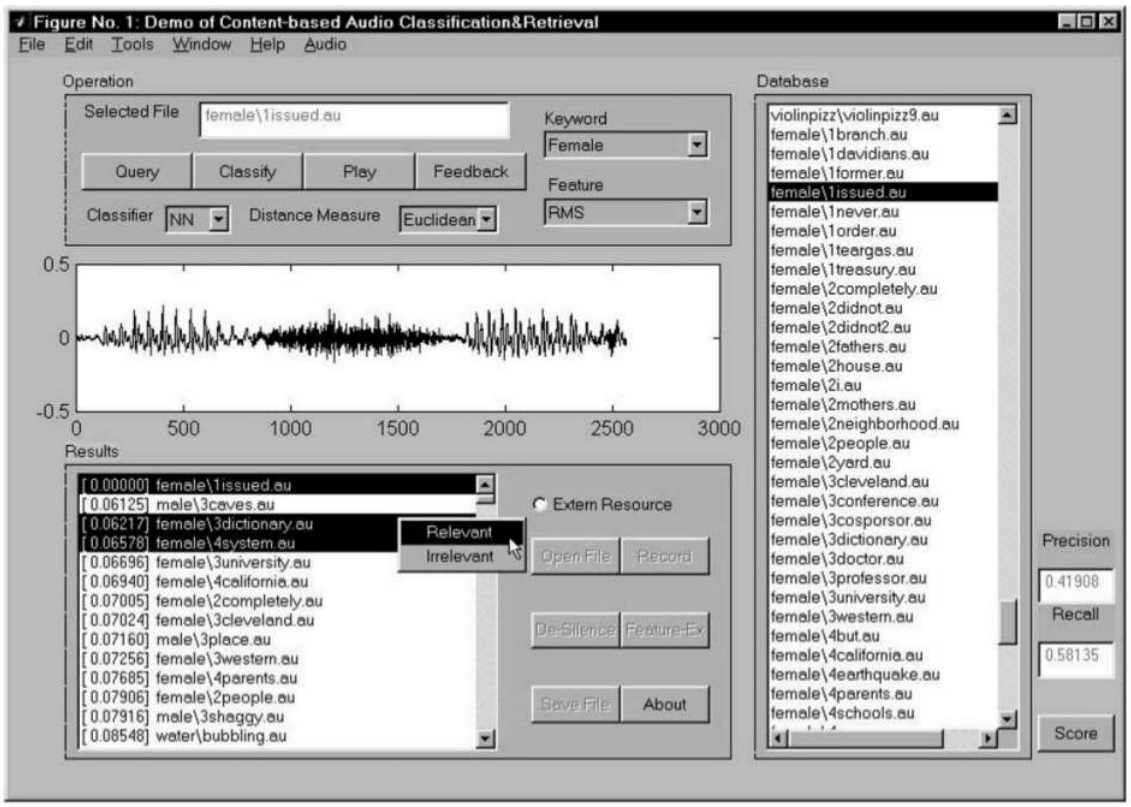


Figure 2.15: The GUI of the audio retrieval system in [68].

In [39, 68], two relevance feedback techniques for retrieval of information in audio domains are presented and discussed. Both algorithms are implemented by updating the weights in the similar distance measurement of audio (e.g. Euclidean distance) between the query and sample audio files and also by updating the query itself, but the second feedback algorithm features the capability of "handling negative feedback under the same architecture as positive feedback". Both techniques could be used in multimedia recommendation systems to obtain accurate results based on (positive and negative) user preferences and listening habits.

Figure 2.15 shows an example of an interface from a system than implements Relevance Feedback for finding similar audio segments. The user simply selects an audio segment from the database, at the right side of the figure, and the interface queries the system for similar results. Next, the result list from the retrieval is presented in the sec-

[15] `http://jango.com/`
[16] `http://grooveshark.com/`

tion at the bottom of the screen and, finally, the user is able to play and listen to the audio tracks, select results and mark them as relevant or irrelevant in relation to to the previous query, regenerating the retrieval list after the feedback is complete.

2.4 Human Computation in Content Classification

People are driven by motivation. Motivation is a psychological feature that arouses an organism to act towards a desired goal and elicits, controls, and sustains certain goal-directed behaviours. In other words, motivation is the purpose or psychological cause of an action.

Human computation systems can be defined as intelligent systems that explicitly organizes human efforts to carry out the process of computation, whether it be performing the basic operations, or taking charge of the control process itself (e.g., specifying what operations need to be performed and in what order). The objective of a human computation system is to find an accurate solution for a pre-specified computational problem in the most efficient way [35].

This section will present the main theories behind human motivation and how is applied to introduce humans into the processing loop of machines by using game elements.

2.4.1 Maslow's Hierarchy of Needs

In 1943, Abraham Maslow presented a theory about human motivation based on his observations of humans, known as the Maslow's Hierarchy of Needs [43]. The hierarchy of needs is portrayed in the shape of a pyramid (figure 2.16), where the needs at the bottom are considered more basic and fundamental than the ones on the top. The theory suggests that an individual must meet the basic level of needs before finding motivation (or desire) to fulfil the next level. The needs are separated in physiological, safety, love and belonging, (self) esteem and self-actualization needs.

Physiological needs are essential and basic needs for human survival, including metabolic requirements such as breathing and food. Safety needs are related to environment stability and feeling safe (e.g. personal, financial and health security). Love and belonging is related to the human need to feel a sense of belonging and acceptance among their social groups, regardless of the group size, including the need to love and to be loved by others. Also, humans need to feel respected by others and themselves, and this is included in the Esteem needs, related to the self-esteem and self respect. People with low self-esteem often will need respect from other individuals and feel the need for fame and glory. Maslow observed that people often engage in a profession or hobby to gain recognition, giving the person a sense of contribution or value. The last group of needs is described as a "desire to accomplish everything that one can, to become the most that one can be". The author portrays a self-actualizer as a person who is living creatively and fully using his

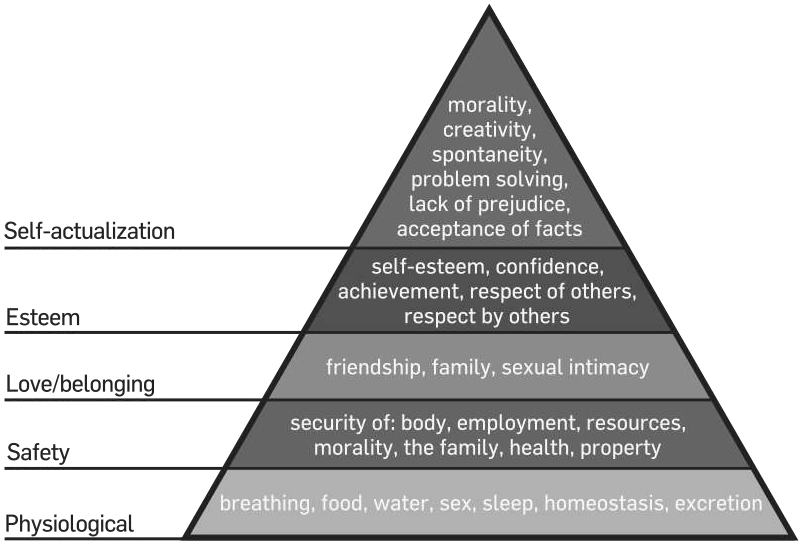


Figure 2.16: An interpretation of Maslow's hierarchy of needs.

or her potentials. Although Mallow's theory has been subject of criticisms, Maslow believes that these needs are what motivates people to do the things they do. It is possible to state that Maslow's Hierarchy of Needs theory is basically the "carrot and the stick approach" of motivation, where human behaviours are basically driven by their desire to satisfy physical and psychological needs.

There are plenty of studies about human motivation and how to generate it towards a specific goal, though

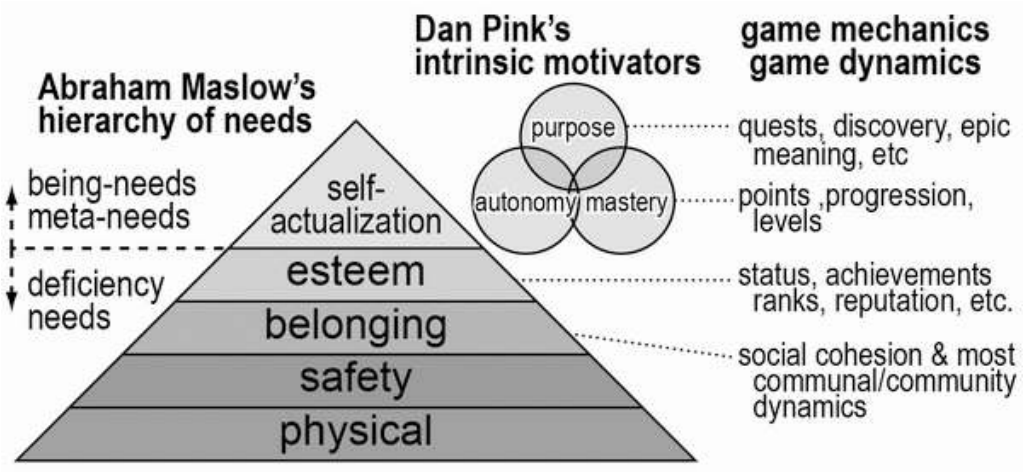


Figure 2.17: Dans Pink's intrinsic motivators.

More recently, Daniel Pink hypothesizes that in the modern society where the lower levels of the Maslow's hierarchy are more or less satisfied, people become more and more motivated by other intrinsic motivators. These intrinsic motivators are precisely the meta-motivators that Maslow is referring to in the self-actualization level, and Pink specifically focuses on three of these: Autonomy, Mastery and Purpose. Many of these needs and

motivators are very similar to game mechanics and dynamics, as it can be seen in figure 2.17.

2.4.2 Skinner Box

On a different branch of psychology, the American psychologist Burrhus Frederic Skinner proposed a Behaviourist approach to define human behaviour as a result of the cumulative effects of environmental reinforcements and learning. He is known by developing an interesting experience called the Operant Conditioning Chamber, also known as Skinner Box.

The Skinner Box was a laboratory apparatus used to study both operand conditioning and classical conditioning [57]. Operant conditioning is a type of learning which the behaviour is modified by its consequences, coined by Skinner in 1937 [56]. This behaviour is initially spontaneous, contrasting a response to a prior stimulus (Classical conditioning), whose consequences may reinforce or inhibit recurrent manifestations of that behaviour. By other words, Operant conditioning means changing of behaviour by the use of reinforcement which is given after the desired response.

Skinner identified three types of responses or operants that can follow behaviour [45]:

- Neutral operants: responses from the environment that neither increase nor decrease the probability of a behaviour being repeated.
- Reinforcers: Responses from the environment that increase the probability of a behaviour being repeated. Reinforcers can be either positive or negative
- Punishers: Response from the environment that decrease the likelihood of a behaviour being repeated. Punishment weakens behaviour.

The Skinner Box was composed by a structure forming a box shaped chamber large enough to accommodate the animal used as a test subject, usually rodents, pigeons and primates. The chamber had at least one switch or lever that could automatically detect the occurrence of a behavioural response or action.

Skinner studied how positive reinforcement worked by placing a hungry rat inside his Skinner Box and rewarding the rat with a food pellet that would be dropped inside the box each time the rat stepped on the lever. The consequence of receiving the food would cause the rat to repeat the action again and again. Skinner observed that positive reinforcement strengthens a behaviour by providing a consequence an individual finds rewarding. Also, he observed that the removal of an unpleasant reinforcer can also strengthen behaviour by submitting the rat to an unpleasant electric current until the rat knocks the lever to switch it off. To study punishers, Skinner would change the reward by activating the switch by a direct unpleasant stimulus, like a electric shock. He observed the opposite of reinforcement, punishers, are designed to weaken or eliminate responses rather than increasing it.

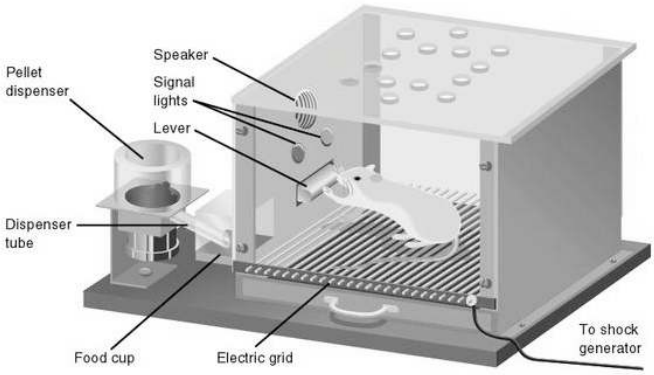


Figure 2.18: Skinner box.

The Skinner Box experiment is directly associated with the motivation behind the realization of repetitive tasks. Many game dynamics have been developed using the principles from Skinner's work, because a point system is often core to many game dynamics, including progression dynamics and levels. With a proper reinforcement schedule, Skinner believes we can ignore people's innate needs and just reward them instead (e.g. with points), and people will learn and be motivated simply by accumulating these rewards.

2.4.3 Csíkszentmihályi's Flow Theory

Flow is considered a mental state of operation in which a person is fully immersed in a feeling of energized focus, full involvement and enjoyment, during the process of an activity, in the theory proposed by Mihály Csíkszentmihályi [9]. According to Csikszentmihalyi, flow is completely focused motivation. It can be attained when the challenges we encounter are matched to our ability.

Figure 2.19 presents the resulting mental states in terms of challenge level (complexity) and skill level (expertise), where the flow area is highlighted. When the task presents to be slightly difficult or easy, we fall out of flow and go into a state of arousal or control about the situation. For situations where the difficulty greatly exceeds our skills, we are likely to fall into an anxiety experience, where the opposite would trigger experiences of boredom.

This also implies that when we are in a state of control or relaxation, we have to challenge ourselves and pick a more difficult task to get back into flow. However, picking a task that is too hard means we must learn and increase our skills gradually in order to move back into flow. Therefore we can infer that Arousal is a zone where we learn the most. Moving info the flow stage is not easy because most tasks people face daily usually do not have a continuous increasing range of difficulty.

The flow model is similar to a more recent model, usually known as the Comfort Zone Model [6, 40] and frequently used in non-formal education and represented in figure 2.20 (left image). Here the Comfort Zone is also considered a type of metal conditioning that creates an unfounded sense of security, which a person may have tendency to contain herself within this zone without stepping outside of it. To step outside of their comfort zone, a person must look for new experiences that may make them feel uncomfortable and unfamiliar situations - the Growth Zone. If the experience is completely different

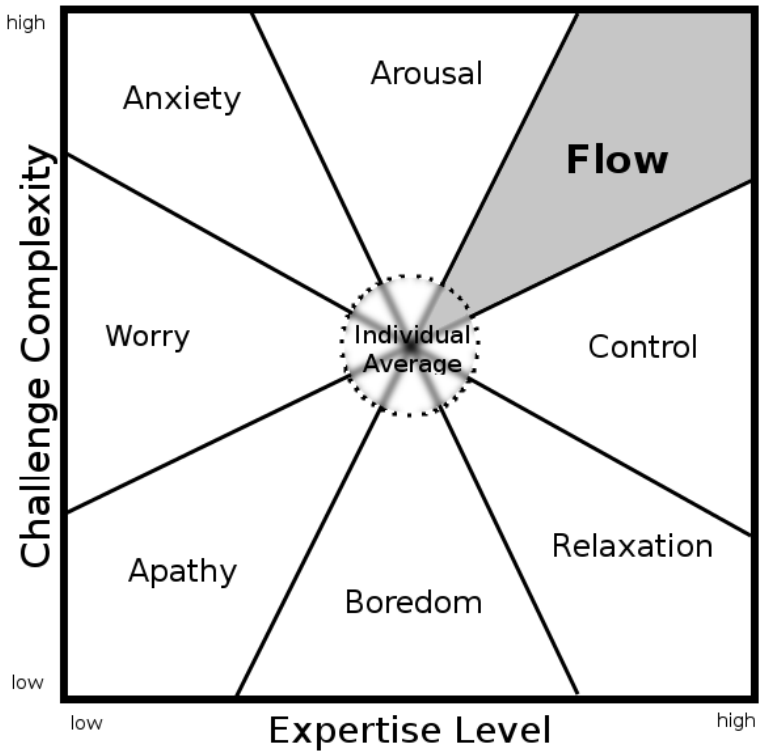


Figure 2.19: Representation of Csikszentmihalyi's flow model: Mental state in terms of challenge level and expertise level, related to a individual average in the centre.

and wraps high complexity problems, the person may experience anxiety and arousal - the Panic Zone - and will require a great focus and learning to return to the grown and comfort zones. Some people may find engagement and joy by keeping themselves inside the Growth Zone.

We can relate the comfort zone model to the flow model by associating the Comfort Zone to the states of Relaxation and Control, the Panic Zone to Anxiety and Arousal, and the Growth Zone to the Flow state.

An area where people experience flow continuously is games. Game developers strive to engage players into tirelessly playing their games for hours, if not days, straight. Games in general can create an artificial environment where the task difficulty is well-controlled and increase gradually. This makes it much easier for gamers to pick a game challenging enough to move them into flow (figure 2.20-right, B2) [70]. If the gamer picks challenging tasks inside the game, it would most likely not be something totally beyond his skill, causing an arousal experience (B3) rather than anxiety (B4), which is undesirable. Again, in arousal state, gamers have to learn to increase their skills sufficiently to move back into flow (C), encouraging gamers to take on more challenges.

The Flow Model is directly associated with states of engagement and immersion expe-

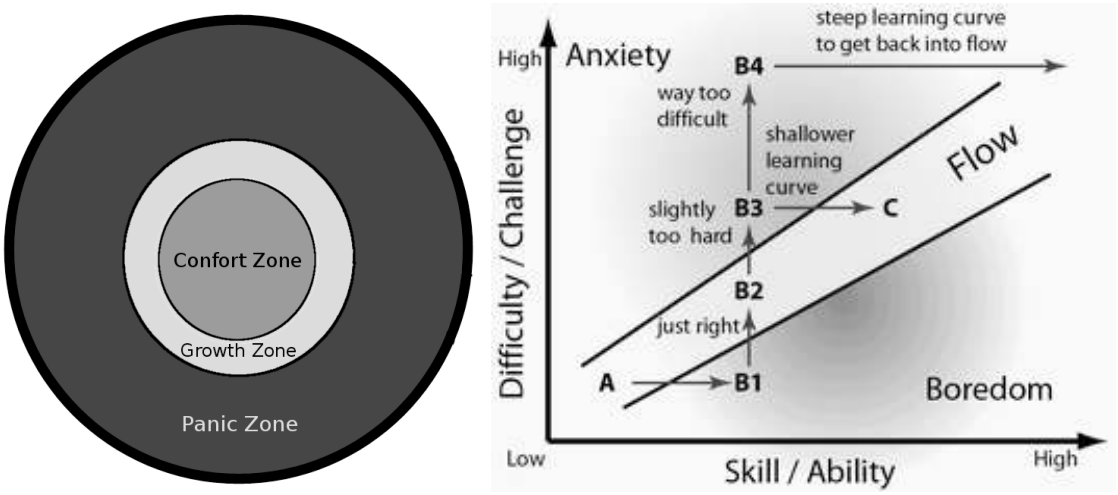


Figure 2.20: Confort Zone Model (left image) and Flow model applied to Games (right image).

rienced by people while carrying specific tasks or playing games, taking in consideration the individual ability and task's complexity, and how these factors must progress continuously to keep activities enjoyable.

2.4.4 Gamification and Human Computation

People are constantly affected by reinforcers in a daily basis. People are motivated to do an action for something in exchange. People work for money, for recognition, for promotions. People study to increase their knowledge, their culture, their technical abilities. Some will study for higher marks, an unfortunate result of an education system focused only on examinations rather than training students for the future and really testing their knowledge, where students are forced to take tests that show only their retention powers, not their actual capacity or knowledge.

Even volunteering activities are developed to obtain some sort of recognition and gratitude that make us feel better and comfortable, or simply by the produced feeling of self-worth and respect. Volunteering is also renowned for skill development, socialization, and fun.

"'Nothing is free except what comes from within you."' - *Robert D Dangoor*

People always receive something in return for their effort, even if is not visible.

For this purpose, humans require some sort of motivation, some mechanism to incite them in repeating the same task over and over, e.g. by using positive reinforcers. A prime example for the use of reinforcers are games. In most game genres, users end up repeating the same group of tasks to achieve a goal and have the sensation of amusement and fun as a player. Here the concept of Gamification is used to create features to compel users to participate in repetitive tasks.

Gamification can be defined as *the use of game design elements in non-game contexts with the purpose of engaging users and solve problems* [11, 12]. Investigators traced the first use of this definition back to 2008 [11], where some consider Brett Terill as the first person using it in a blog post [27].

It is possible to use game elements to augment and complement the entertaining quality of a diverse range of systems. Some properties are considered important for applying such elements: *Persuasion* and *Motivation* to induce and facilitate the participation of users in a specific task; *Engagement, Joy*, and *Fun* to compel users to spend more time on the task, even without realizing it; *Reward* and *Reputation* to induce a purpose and therefore the users participation.

In [27], Gamification is defined as: "*a process of enhancing a service with affordances for gameful experiences in order to support user's overall value creation*". The author refers to game design elements as services and games as service systems, and argues that the experience of playing a game is deeply individual, meaning that each service has its own value uniquely determined by the user's subjective experience. Therefore, the use of game design elements may lead to a gameful experience with one user but not with another one. This definition supports the idea that gamification cannot be based on a set of methods or mechanics but as a process where the designer attempts to increase the chance for users to have gameful experiences through adding a set of qualities to the service system. As such, the process of gamification does not need to be necessarily successful.

Gamification is deeply related to motivation theories proposed by Abraham Maslow, Dan Pink, Burrhus Skinner and Mihály Csíkszentmihályi: Maslow and Pink's theories explains how people motivation are driven by their innate and by detailing and classifying such needs. Pink's theory also extends Maslow's hierarchy by adding motivators very similar to game mechanics and dynamics (see figure 2.17). On the other hand, Skinner believes that under a proper reinforcement schedule, we can ignore these innate needs and just give points to people instead, to motivate them. However, using Csíkszentmihályi's flow theory, we observe that should not give blindly points to people, without putting some sense of purpose in it. Gamification requires a constant adaptation to people's skill by searching for the correct amount of complexity/uncertainty to keep them engaged for long time periods, achieving the state of Flow described by Csíkszentmihályi - a state of optimal intrinsic motivation.

There are some projects and scientific articles describing systems and application of game elements in different forms of multimedia. A significant approach tries to solve the cold-start problem of automated data modelling used in the recognition and automatic classification of multimedia content. Most resorting to crowd-sourcing approaches for greatly improving the collection of classification data.

Projects which use game elements as a tool for engaging users into content classifica-

tion of images and audio are going to be presented and analysed in the following sections.

2.4.5 ESP Game

Example of Human Computation through a gamification approach for engaging users into content classification, the ESP Game[17] [64] is always referred to as a good one. It is a two player game for the purpose of labelling images with textual information. This concept was used later as an inspiration for the creation of the Google Image Labeller game [29].

The generated labels by the ESP could be applied in a variety of applications, such as for accessibility purposes for visually impaired individuals and elderly people, or in large databases of labelled images for machine learning algorithms.

Figure 2.21: The ESP Game. Players try to "agree" on as many images as they can in 2.5 minutes. The thermometer at the bottom measures how many images partners have agreed on [64].

At least two players are required for playing the game, which assigns randomly the same images through each game session. Ideally, players do not know each other and must not have any kind of contact between each other while playing the game. The purpose of the ESP game is to guess what your partner is typing for each image which both can see. Once both type the same string, the game moves on to a new image.

The game uses an output-agreement approach. Points are rewarded when players reach an agreement for each image, achieving extra points for agreement on a large number of consecutive images. This bonus is represented by the thermometer at the bottom of

[17] ESP stands for Extra Sensory Perception.

the screen in figure 2.21.

To step up the game difficulty and improve the range of results for each image, the game generates a list of taboo words for each image. These words are obtained from the game itself and are introduced to force players to use less obvious terms and provide a richer description of the image. Players are not allowed to type any of the taboo words, nor their singular and plural forms, or phrases containing those words.

Usually, players loose interest once a game is beaten either by mastery or through cheating. The game is meant to be played by hundreds or thousands of players online at the same time, from sparse locations around the world. Players are teamed up with random players, reducing the probability of these being partnered with themselves or people from the same location. To minimize cheating, the ESP game enforces the requirement of each partner having different IP addresses and uses pre-recorded gameplay to prevent a massive agreement strategy. Massive global agreement of a strategy can be detected by monitoring changes in the average time in which players are agreeing on images. Enforcing taboo words across entire play sessions would also work as prevention mechanisms for agreement strategies (e.g. if the user typed 'a', this word would stay marked as taboo for the rest of the session).

2.4.6 Peekaboom

With a purpose similar to the ESP Game, Peekaboom [66] is another output-agreement multi-player game for locating objects inside images.

The game divides the gameplay in two simple roles: "Peek" and "Boom". Random players are teamed up in pairs and each one takes the role of Peek and Boom. The goal of the game is for Boom to reveal parts of the image to Peek, so that Peek can guess the associated word. Figure 2.22 displays the differences in the user interface for both roles of Peek (in the left) and Boom (in the right).

Each image is paired with an associated word which only Boom can see. These words are obtained from the ESP Game. For this purpose, Boom can click in the image to reveal to Peek the parts associated to the word, while Peek has to enter guesses for the word associated to the image. To receive more points, Boom has an incentive to reveal only small areas of the image enough for Peek to guess the correct word. Each click reveals a 20 pixel radius area around the cursor. When Peek correctly guesses the word, the players are rewarded with points and proceed to a new image-word pair by switching roles. The game provides the option to skip the current image if it is hard, where players receive no points. To disambiguate the content of some images, Boom can use a feature called "pings" to point and signal locations on Peek screen inside the image, by simply right clicking in the image. Another feature of the game involves hint buttons which allows Boom to give hints to Peek about how the word relates to the image (i.e. is a noun, a related noun, a verb, etc.).

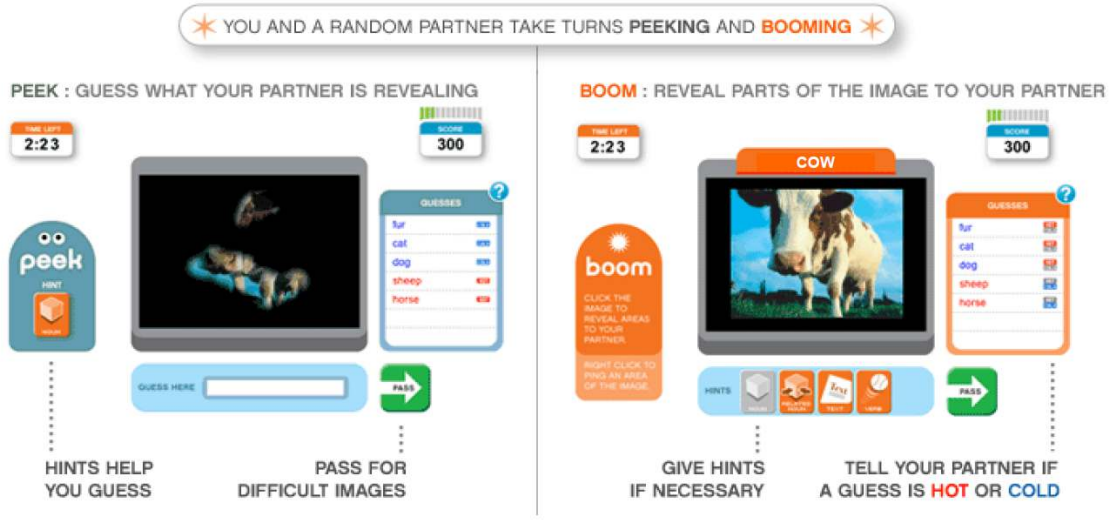


Figure 2.22: The Peekaboom Game interface for both roles of Peek (left) and Boom (right).

The purpose of the game is to construct a database for training computer vision algorithms and eliminate existing poor image-word pairs. Peekaboom collects information about how words relate to the image, the pixels necessary to guess a word allowing to determine what kind of object a set of pixels constitute, and the pixels inside the object. The image-word pairs could be considered poor by observing the number of players who agree to skip an image without taking any action on it, because these images are likely to be hard due to a poor relationship between the associated words (the quality of the image is not good or the relationship is simply strange and questionable).

2.4.7 TagaTune

TagaTune is a game inspired by and similar to the ESP game, which uses a human computation paradigm for gathering perceptually and meaningful descriptions for audio data based on a output-agreement approach.

Players are paired to classify audio tracks and at the same time attempting to guess whatever their partners are thinking.

The main difference between TagaTune and the the ESP game is the classification of small audio tracks instead of images, resulting in an increased diversity of labels due to the increased subjectiveness and ambiguity of the audio listening (e.g. different sources can produce similar noises, music tracks can be abstract and ambiguous, etc.). Figure 2.23 displays the prototype's interface. Part of the labelled data is based on shared perception of the sounds and has the potential of improving CAPTCHA[18] accessibility and can be

[18] "Completely Automated Public Turing test to tell Computers and Humans Apart", a type of challenge-response test used in computing to determine whether or not the user is human [63], used frequently over the internet.

useful in psychoacoustics or phenomenological research.

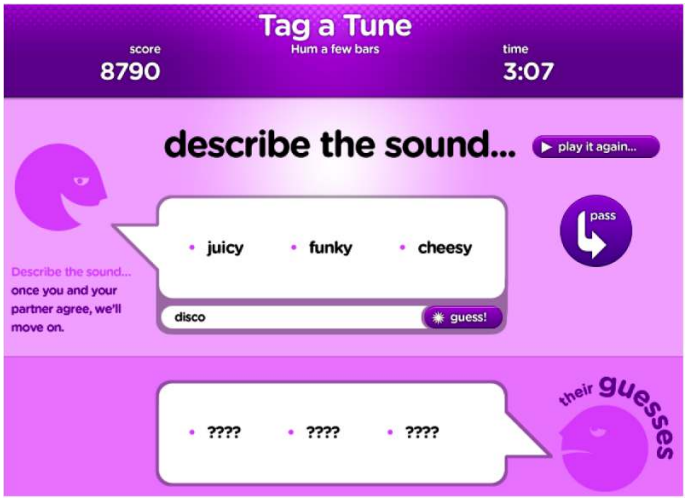


Figure 2.23: Preliminary interface for the TagaTune prototype.

2.4.8 MoodSwings

The MoodSwings game [46] aims to label songs according to their mood. It is a collaborative game playable by two players where users are asked to indicate the mood by clicking on a two dimensional plane showing the traditional valence-arousal axes, where valence, reflecting positive vs. negative emotion, is displayed on the vertical axis, and arousal, reflecting emotional intensity, is displayed on the horizontal axis. Players are selected and paired randomly and anonymously over the internet, with the objective of reaching agreement on the mood of an audio piece drawn from a music database.

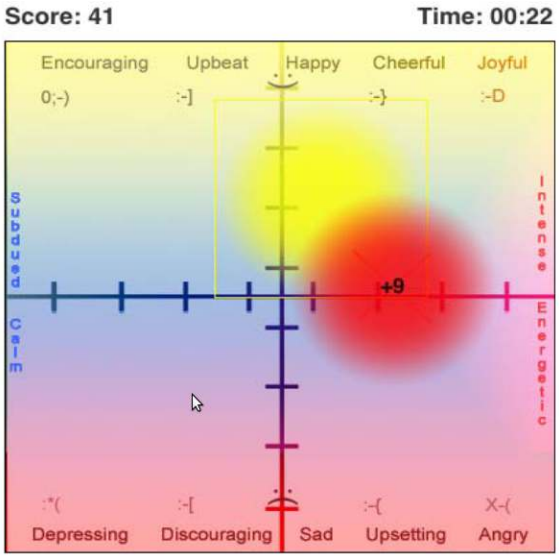


Figure 2.24: Moodswings gameplay. Red and yellow orbs represent each player.

Scores are given as rewards to the players and are based on the amount of congruency between the two cursor positions. Players earn more points by overlapping their partner's cursor, but they are visible only once every 3 seconds. However, cursor size decreases over time increasing the difficulty of overlapping cursor positions.

The originality of the approach is that tagging occurs in a continuous way while players interact. MoodSwings is also an output-agreement game because reward is based on the matching of players valence/arousal indications.

2.4.9 Other Related Work

There are other related projects which have been explored. Most of them were used as inspiration or are presented as related work of the projects presented above in this section, hence the lack of noteworthy differences.

Verbosity [65] is a web-based "output-agreement" game to gather common facts. The game is played by two players that cannot communicate. HerdIt [4] originality is based on being deployed on Facebook platform, benefiting from the Social Networking effect. MajorMiner [41] is a single player game based on the output-agreement paradigm that asks users to assign labels to music clips in a asynchronous way, i.e. users are not required to be online at the same time to play the game. The ListenGame [60] is also meant to label music, but here players choose among a predefined list of labels those that fit best and worse a given music piece, being able to compute a confidence level associated to each tag based on the answers of several players.

2.5 Web Technologies and Libraries

Internet is considered one of the biggest media used for communication in worldwide scale and keeps evolving fast. Statistics for the year 2012 report an approximated total of 634 million of websites (of which 51 million were added in the same year) present in the internet [48], 256 million of domain name registrations across all top level domains [62] and 2.4 billion users available worldwide [72]. An approximated number of 144 billion emails were sent worldwide by a total of 2,2 billion email users [54]. These values correspond to what it is known as the *visible Web*[19], a portion of the world wide web that is indexable by search engines, but there is more information that is not reachable by search crawlers, therefore not covered by statistics. This justifies the interest in multimedia classification systems based on content, and similar to SoundsLike (chapter 4) to enable automatic classification of these web contents, unreachable by search crawlers.

In this section I will summarize some web technologies used in the development of web applications and relevant for the work performed during the project.

[19] Also known as *surface Web*, *Clearnet*, or *indexable Web*

2.5.1 HTML and HTML 5

HTML (HyperText Markup Language) is a markup language widely used over the internet to create web pages. The language is composed by elements that consists on tags enclosed in angle brackets, within the page contents. HTML documents are meant to be interpreted by a web browser to compose visible or audible web pages.

HTML can be used to create structured documents by defining structural semantics such as paragraphs, lists, quotes and links. It is also possible to add images, create forms and embed scripts written in others languages such as JavaScript. HTML itself is limited and static, requiring others technologies for the creation of full interactive interfaces. Cascading Style Sheets (CSS) is a language used to define appearance and layout of the HTML content, while Javascript is commonly use to define behaviour and manipulate the HTML elements and content. Both languages can be embedded directly inside HTML documents or linked through external files. Due to the importance of CSS and JavaScript, they are described in separated sections.

The HTML standard was created in 1990, where the most used version has been HTML 4.0, standardized in 1997 [33].

HTML5 is a new revision of the HTML standard which is intended to replace and improve HTML4 and XHTML. It includes detailed processing models to encourage more interoperable implementations, introducing programming interfaces for complex web application and cross-platform mobile applications. HTML5 adds new features such as multimedia support by video, audio and canvas elements, the integration of scalable vector graphics (SVG) content, new elements to enrich the semantic content of documents (such as section, article, header, etc.), client offline data caching and storage, and more.

2.5.2 CSS

CSS is a language used to define the appearance and disposition of elements in a markup language. It was designed to separate the document content from document presentation, aiming to improve content accessibility, providing more flexibility on presentation characteristics and repetition in structural content (e.g. multiple style-sheets) [67]. It allows to define different styles for different rendering methods such as on-screen, print, Braille based or tactile devices, etc. It also allows the definition of style rules that depend on the screen size or device on which it is being viewed. The most current version of the standard is CSS 3, which divides the entire specification in several separated documents called “modules”. Each module adds new capabilities or extends features defined in CSS 2, preserving backward compatibility.

2.5.3 Javascript

Javascript is an interpreted computer programming language, formalized in the ECMAScript language, primarily used by web browsers to allow the implementation of client side scripts for controlling web applications behaviours and improving user interaction [18]. It is used for document content manipulations, animations and asynchronous communications, and also used in server-side programming, game development, and creation of desktop and mobile applications.

In this section, some relevant libraries available in JavaScript will be presented.

2.5.3.1 jQuery

jQuery[20] is a free and open source JavaScript library designed to simplify the client-side scripting of web applications. It provides features to make it easier to navigate a document structure, select DOM elements, handle events, create and control animations and develop asynchronous web (AJAX) application.

It aims to be cross-browser, support new web technologies, including CSS 3 selectors, and have a lightweight footprint. jQuery allows the creation of abstractions for low-level interaction and animations, advanced interface effects and high level, theme-able widgets. It uses a modular approach that allows the addition of new features for the creation of powerful dynamic web pages and web applications.

As a popular sub-project of jQuery, jQuery UI is a set of user interface interactions, advanced effects, widgets and themes built on top of the jQuery library aimed to be used by developers and designers in the construction of highly interactive web applications.

2.5.3.2 D3.js

D3.js[21], acronym for Data-Driven Documents, is a JavaScript library for manipulating documents based on data, combining powerful visualization components and a data-driven approach to DOM manipulation. With this library, it is possible the creation of multiple visualizations using simple HTML or Scalable Vector Graphics (SVG) elements inside a canvas area (this canvas is a simple HTML div or SVG canvas, **not** a HTML 5 canvas).

The library gives the developer the possibility of choosing the use of simple SVG primitive elements or HTML elements visualization creation. Both methods can be combined are similar and widely supported by Web Browsers, where the main difference between them are the different properties each element offers to the programmer.

[20] jQuery home: `http://jquery.com/`
[21] D3js home: `http://d3js.org/`

2.5.3.3 Raphaël

Raphaël[22] is a JavaScript library aiming to simplify developers work with vector graphics across all web browsers.The library uses the SVG W3C Recommendation and VML (Vector Markup Language) as a base for creating graphics. This means every graphical object created is also a DOM object, making possible to attach JavaScript event handlers or to modify them later.

2.5.3.4 Buzz!

Buzz![23] is a Javascript library that allows developers to easily take advantage of the new HTML5 audio element through a well documented API. Due to HTML5 being fairly new and an unfinished standard, the library degrades silently on non-modern browsers.

2.5.3.5 Node.js

Node.js[24] is a software platform used to build applications using JavaScript as programming language. It contains a built-in HTTP server library, being commonly used for server-side applications and achieves a high throughput via non-blocking Input/Output and a single-threaded event loop. It is attractive to web developers due to the fact of being able to create client and server-side applications without the need of learning other programming language besides JavaScript.

2.5.3.6 Modernizr

Modernizr[25] is a small JavaScript library with the purpose of detecting the availability of native implementations for new web technologies implemented in the HTML5 and CSS 3 specifications.

It uses feature detection, rather than identifying the browser user agent. This approach is more reliable since the same rendering engine could not support the same feature in different platforms, and the user-agent and platform can be changed by the user, usually used as a workaround to unblock features disabled by web applications for certain platforms or user agents.

The library simply tests the platform for a list of features available and then creates a JavaScript object named “Modernizr” that contains the results of these tests as boolean properties. It also adds classes to the HTML element based on what features are and are not natively supported.

[22]Raphaël home: `http://raphaeljs.com/`
[23]Buzz! home: `http://buzz.jaysalvat.com/`
[24]Nodejs home: `http://nodejs.org/`
[25]Modernizr home: `http://modernizr.com/`

2.5.4 PHP

PHP[26] is a free programming language designed mainly for web development and server-side applications. The code can be embedded directly into an HTML source document, which is interpreted by a webserver with a processor module, generating the result web page that is sent to the client. PHP can be used also as a command-line program and can be used for creating standalone graphical applications.

Created in 1995 by Rasmus Lerdorf, PHP originally stood for Personal Home Page, but now is a recursive acronym of PHP Hypertex Preprocessor [73].

2.5.5 REST

Representational state transfer, or REST, is an architectural style for a remote procedure call that conventionally is defined by clients and servers, where clients initiate requests to servers and obtain a response after the process of the request by the server. REST relies on the semantics of the HTTP protocol. Requests and responses are built around the transfer of representation of resources. A representation of a resource can be a document that provides the current or intended state of a resource. The representation of each application state may contain links or tokens that may be used the next time a client chooses to initiate a new request.

The client begins by sending requests when it is ready to make the transition to a new state. While one or more requests are outstanding, the client is considered to be in transition. The representation of each application state contains links that may be used the next time the client chooses to initiate a new state-transition [1, 17].

Here is a list of some of the key goals REST includes:

- Scalability of component interactions;
- Generality of interfaces;
- Independent deployment of components;
- Intermediary components to reduce latency, enforce security and encapsulate legacy systems.

The REST architectural style describes six constraints [17] applied to the architecture:

- Separation of Concerns (Client/Server) - A uniform interface is used as a separation of concerts between the clients and servers. This interface simplifies and decouples the architecture. Through this separation, the portability of the code and the decoupling of both the client and server is possible. Clients will not be concerned about the implementation behind the server interface. Servers are not concerned about the user state, increasing scalability.

[26] PHP home: `http://www.php.net/`

- Uniform interface - The uniform interface between client and servers simplifies and decouples the architecture.
- Stateless - No client context is stored on the server between requests. Each request from any client must contain all the necessary information to service the request. Session states are also held in the client.
- Caching - Responses must implicitly or explicitly define themselves as cacheable or not by the client. A good caching policy will eliminate some useless client-server interactions.
- Layered system - Communication between client and the end server must support the introduction of transparent intermediary servers, and clients cannot determine if they are directly connected to the end server. This intermediary servers may provide stability by adding load-balancing, validation or caching of requests.

2.6 Previous Work

In this section, previous work in the area is going to be presented, integrated in the VIRUS: Video Information Retrieval Using Subtitles research project, developed in the context of the HCIM: Human-Computer Interaction and Multimedia Research Team group, the project where this thesis was developed around.

2.6.1 iFelt

iFelt is an interactive web application for cataloguing, exploring, visualizing and accessing the emotional information of movies from the audience perspective [51]. It was written in Flash with the purpose of exploring the emotional dimension of movies, according to its properties and users emotional impact profiles, preferences and moods.

Two great challenges addressed in this project include:

1. Classification of emotional content of movies by offering a video classification based on emotions mainly felt by the users, to be related with emotions expressed in the movies.

2. Access and exploration based on the movie emotional properties and by the user profile and current mood.

The emotion classification uses a tagging system based on the Ekman six Basic Emotion Model [14], the most consensual emotional model according to the authors. These tags include the six emotions: happiness, sadness, surprise, fear, anger and disgust. This classification model was also applied in the detection and classification of the users emotions from physiologic patterns (heart rate, respiration and galvanic skin conductance) based on a process that is being developed, with good results, in spite of the

challenges [50] and identifying all but the surprise emotion. Plutchik emotional model (1980) is also used for applying colours to the emotions representations.

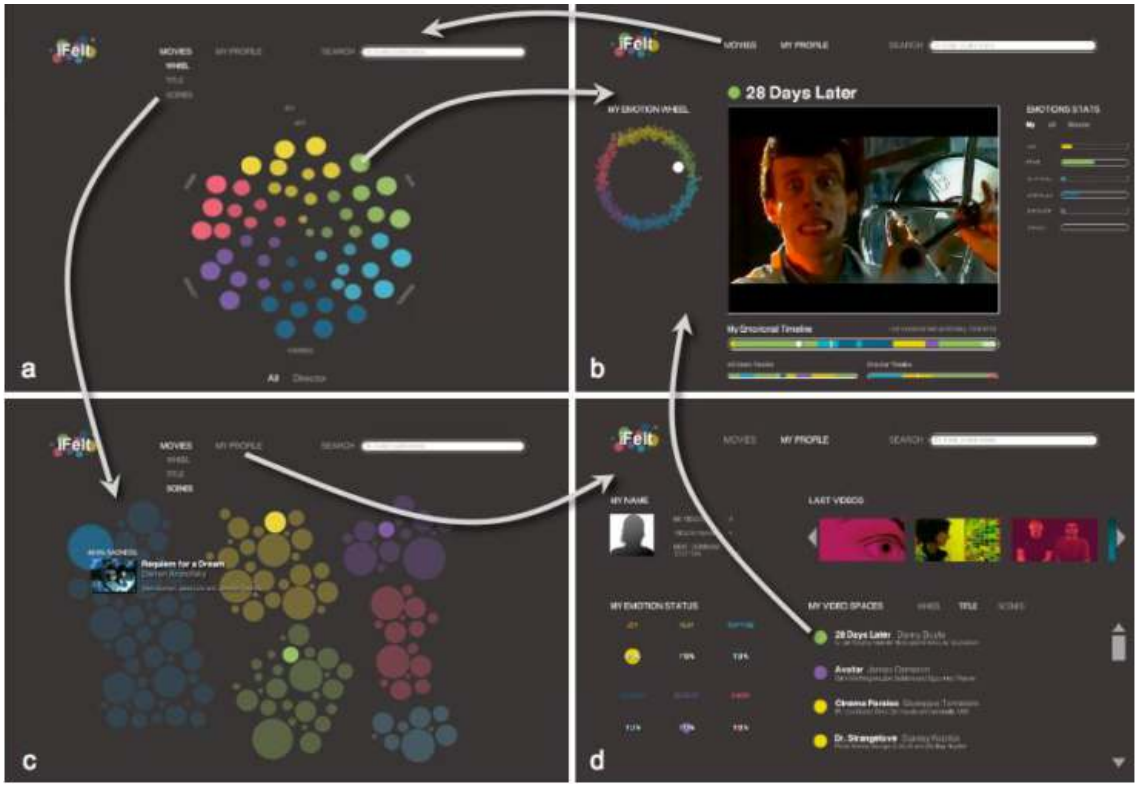


Figure 2.25: iFelt: a) Movie Spaces; b) Movie Emotional Scenes Space; c) Movie Emotional Profile; d) User Emotion Profile.

There are four important perspectives in iFelt that can be observed in figure 2.25:

1. Movie Spaces: here, the users have an overview of movies that may be preselected by a query in iFelt with information about their dominant emotions. They can browse to any of the displayed movies and watch them individually. There are two possible representations:

   (a) Movies Emotional Wheel: in this perspective, a user can quickly browse and select movies that are organized by their dominant emotion in a circular grid. The distance to the center represents the level of dominance for that emotion.

   (b) Movies Title List: this perspective displays the movies in a list. Each entry has an image, the film's title and a small coloured circle representing the dominant emotion of that movie.

2. Movies Emotional Scenes Space: here, users have access to an overview of scenes from movies, based on their dominant emotions. Scenes are represented with coloured circles representing their dominant emotions, where the circle size relates to the dominance level. Hovering over the circles displays details about the scenes

of the respective movie. Users are also able to select scenes for a specific movie and emotion, and obtain an emotional summary of the movie from the selected scenes.

3. Movie Emotional Profile: the view where movies can be watched. The most dominant emotion is represented at the top of the video by a coloured circle. There are also available bar graphs, at the right, displaying the dominant emotions by their percentage in the movie, and, at the bottom, three emotional timelines displaying the felt emotions through a temporal dimension, along the movie:

   (a) My Emotional Timeline, where the user is able to observe the classification of the movie based of his felt emotions while watching it.

   (b) All Users Timeline, where the system processes and displays the average of the emotions felt by all users for each video.

   (c) The Directors Timeline, where is observable the emotions the video director was expecting to trigger in the audience along the movie.

   Finally, there is a circle of emotions in the left, representing the current emotion as an animated white dot that approaches the felt emotion on the circle as the video plays.

4. User Emotion Profile: here, users can visualize their personal information, browse and obtain overviews about the percentage of felt emotions during the movies previously classified by them.

A user evaluation of iFelt had very satisfactory results and concluded that it was perceived as easy to use and useful and very satisfactory, especially in the exploration of emotional timelines and the ability to compare the users own emotional classifications with directors' and all users' perspectives.

2.6.2 Movie Clouds

Movies Clouds [20] is a Web based application that uses the paradigm of tag clouds in the visualizations and browsing of movies, enabling the analysis and exploration of movies through their content, especially the movie sound effects and subtitles and emphasizing all emotional dimensions such as the ones expressed in the video or the ones felt by the viewer.

The prototype was created using HTML 5 + CSS 3, Javascript and PHP, a set of open web technologies that allows the application to run seamless in the recent updated versions of all popular browsers.

There are two main views in the application: A view for visualizing the video space and another to visualize a specific video with overviews of its contents and five track timelines.

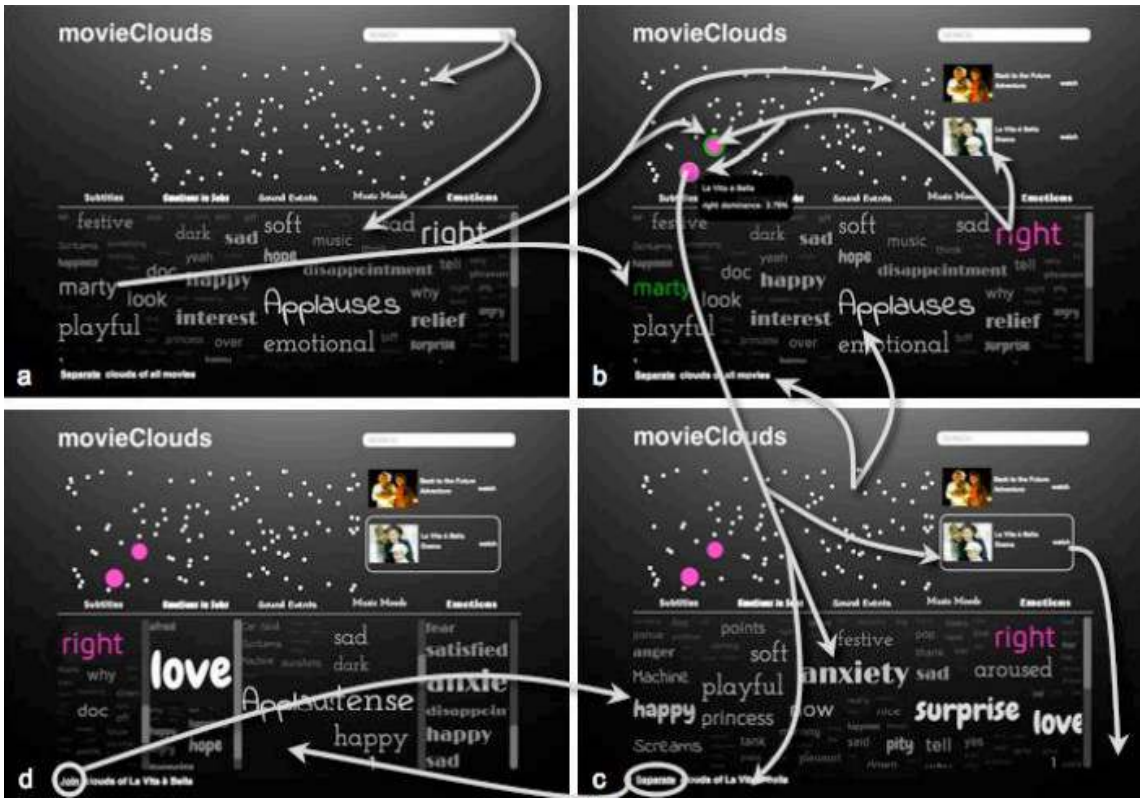


Figure 2.26: MovieClouds Movies Space view navigation

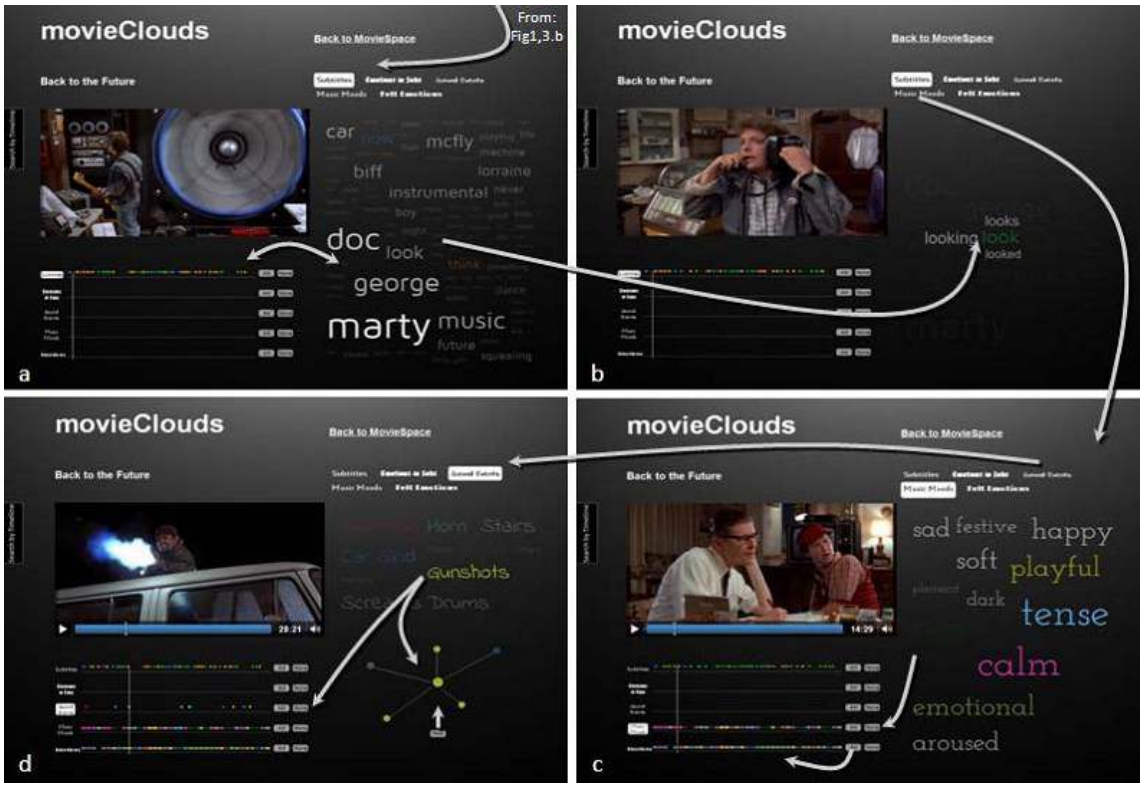


Figure 2.27: MovieClouds Movie view navigation

The "MovieSpaces View" (figure 2.26) allows the search and exploration of the movies present in the MovieClouds database. The view presents the user with a particle cloud of movies and allows the selection of tags with the objective of highlighting the movies in the cloud that matches the dominance of that tag in the contents. The highlight has the exact same colour of the tag, after the selection. When hovering with the mouse over a movie, in the cloud, it is displayed a small information box with generic information about the film such as its title, actors, director, etc. There is also a small list at the right of the cloud presenting information about the selected movie. The tags are divided in 5 tracks: subtitles, emotions in subtitles, sound events, audio moods, and felt emotions. After selecting a movie, the user is redirected to the Movie View.

In the "Movie View" (figure 2.27) the user can visualize the video and observe the tag clouds and track timeline of all 5 categories of tags referred to in the previous view. The position in the timeline represents the playback position in the movie. Highlighting a tag will mark the occurrences (or events) of that tag in the timeline as a circle of the same colour as the highlighted tag. Clicking in a circle in the timeline will start the video playback in that specific time frame, emphasizing the current tag. During the video playback, all tags associated to the current time-frame will be emphasized in the tag cloud (the tag appears brighter than the others). The highlight of more than one tag is possible. The music mood and felt emotions categories are marked with rectangles in the timeline instead of circles, because they may last longer.

The whole application was evaluated about usability and utility, showing interesting and positive results that incited the continuation of the work in the future.

To present rich information about movies, in multiple perspectives, detailed data about the content is necessary. We require a way to collect the necessary data to describe the contents for each movie inside the movie space. This master thesis was developed as a continuation of MovieClouds, by contributing with new representations for movies and audio similarity, with emphasis on SoundsLike, a Game with a purpose that aims to engage users in movies soundtrack labelling, integrated in MovieClouds as a solution to the data collection problem.

Chapter 3

Movie Visualizations

This chapter presents some of the work developed during this thesis to extend functionalities and add new features to MovieClouds. The contributions are divided into two main section: a similarity audio representation to display differences between the content of audio excerpts, allowing soundtrack navigation and help finding similar excerpts, and alternative visualizations to display changes in movies along time and in multiple perspectives.

3.1 Sound Similarity

In MovieClouds it is possible to browse movies by moods and events in audio. However the user is unable to look up for similar audio quickly of they were not previously toggle. To address this issue, a visualization has been developed as a helping tool for the classification of audio excerpts by quickly displaying and allowing to browse similar sounds excerpts.

The visualization would integrate the "Movie View" of MovieClouds.

3.1.1 Motivation

Lets suppose we have a random collection of audio excerpts, which some may be similar in terms of context or are produced from the same source. The task of selecting audios that sound similar may be easy for small collections of sounds excerpts, but for massive collections (e.g. in movies's soundtracks and music library) the task would be very difficult without the help of an automatized audio processing tool. We face the challenge to look up for a representation to display audio differences (or similarities) using visual paradigms that help finding similar audios - given any audio excerpt in a movie soudtrack, in the context of the movie when it appears. This could be useful for movie navigation and analysis, and also as a tool to support audio classification. For this purpose would rely on a previous process to find these differences between audio excerpts.

From this, some questions rise up:

- How we are going to represent audio similarities or differences to the user?
- What paradigms will make users perceive these difference?
- How to focus on the most relevant audio excerpts in a huge amount of audio?
- How to access the audio excerpts in the context of the movies where they belong?

These questions are going to be addressed in the following sections.

3.1.2 Requirements

After some brainstorm sessions, it was decided to represent the differences visually using distance as a measure of the difference between audio contents. The distance between two audio pieces that sounds similar, in terms of content (e.g. two whispers), will be short compared to the distance of audios from distinct events (e.g. silence vs an explosion). Also for each audio excerpt, the most relevant excerpts to compare to would be the closest ones, allowing to cut down the complexity inherent in a huge amount of information.

For this purpose, the representation should display a group of closest audio excerpts and all the similarity relationships between the selected excerpt and its closest neighbours. For which we chose a graph.

The following functional requirements were taken into account when designing the representation:

- It must be possible to select audio excerpts and relations between them;
- The current selected audio excerpt must be highlighted from the remaining excerpts in the visualization;
- It must be possible to listen to a preview of an audio excerpt by a simple interaction with its representation;
- The user must be able to compare similarities by looking at the distances between excerpts and be able to check their sounds;
- It must be possible to select another excerpt present in the visualization to be the new current excerpt.
- It must be interactive, providing a visible and audible feedback in the interactions;
- It must be possible to get context information to identify the audio excerpt in the movie where it belongs to, and use it as a means to navigate the movies based on its audio content.

The following non-functional requirements were addressed:

- The visualization must present the data right away;
- It must be perceived as useful, satisfactory and easy to use;
- The visualization and animations should be smooth and pleasant;
- It must motivate users to participate continuously, even on repetitive tasks;

These requirements will inform the design development and evaluation of the interactive sound similarity visualization.

3.1.3 Design

Sketches were created for the integration of Sound Similarity visualization in the user interface of the current project's prototype, with the objective of exploring of different designs to obtain the best utility and user interaction. From those sketches, some *mockups* were created and discussed with the team, and an HTML prototype was developed.

This visualization, named as Sound Similarity Graph and later integrated in SoundsLike [21], is includes a force-directed graph representing the similarity relation of a selected audio excerpt to the most similar audio excerpts in the movie. Here audio excerpts are represented as nodes and the similarity relation as connections (edges) between nodes. The nodes repel each other and the graph may be dragged around with an elastic behaviour. The similarity is translated to the graph metaphor as the screen distance between the central (current) node and its neighbours nodes (the most similar in the movie). Also the links colours are used to express the distance, were warmer colours express the most similar sounds and cold colours express the most distant nodes.

Figure 3.1 displays a mockup showing how the sound similarity graph segments visualization is going to be integrated into the "Movie View" of the VIRUS Project's MovieClouds prototype. The smaller visualization section in this view has the same functionalities as the maximized version.

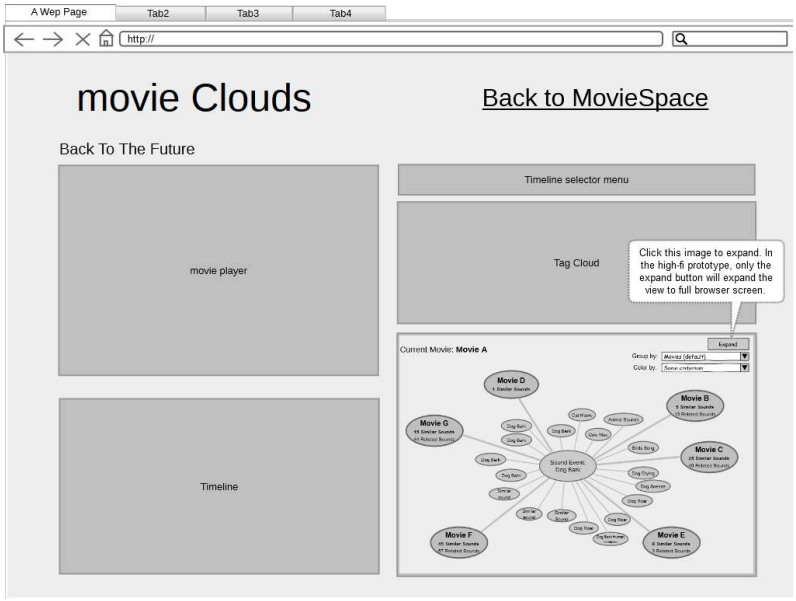


Figure 3.1: *Mockup* of the Sound Similarity Graph visualization into the "Movie View" of the MovieClouds prototype.

Figure 3.2 represents a visualization of the Sound Similarity Graph relative to the central one. The central node represents an audio segment selected for comparison with others, that belongs to a specific film or episode. The smaller nodes are a representation of audio segments that are considered most similar to the one that are connected - the central

node. The bigger nodes are an aggregation of similar audio segments belonging to another film, which a user can click to expand into smaller nodes. Clicking on the smaller nodes or the central one will play the associated sound and show additional information about the segment. There is also the possibility of selecting a new central node and update the visualization to display similar sound segments related to the sound represented by this new central node.

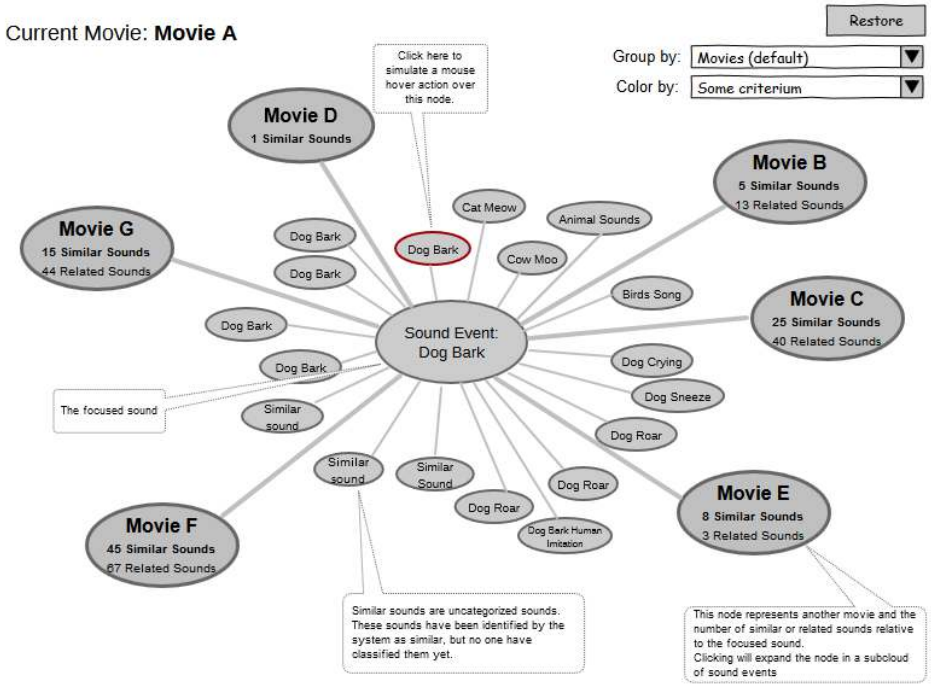


Figure 3.2: *Mockups* for the visualization of Similar Audio Events Prototype, using a "force-directed graph".

3.1.4 The Implementation

From the requirements and designs, a prototype was developed.

Before advancing on implementation details, lets define "audio similarity" value between two audio pieces is based on the comparison of their contents, being the Euclidean distance between audio histograms computed from extracted features (more details in [20]). Two audio pieces are equal if the raw similarity value is 0. A higher value represents a higher difference in terms of audio content.

The graph defines a configurable minimum and maximum screen distances for its edges and normalizes all similarity values between those distances. Since there is no limit on how high the similarity value can achieve (note than 0 means equal), a normalization of values is required for representing the distances on the screen space available for the visualization.

The normalization function f_n displayed in table 3.1 maps the domain $[0, V_{max}]$ to a value between m and M. It is the output of a function C that receives as parameters $m \geq 0$ as the minimum screen distance, $M \geq m$ as the maximum screen distance, and $V_{max} \geq 0$ as the maximum similarity value of the dataset to be represented. V_n is the output of f_N which receives V as parameter and outputs the normalized value V_n.

$$f_N : [0, V_{max}] \mapsto [m, M]$$

$$f_N = C(m, M, V_{max}) : m > 0, M \geq m, V_{max} \geq 0$$

$$V_n = f_N(V) : 0 \leq V \leq V_{max}$$

Table 3.1: Normalization function for audio similarity values.

For the development of a prototype, it was decided to use the same Web technologies already used before in MovieClouds. For this purpose, HTML5, CSS3 and Javascript were used, allowing the easy integration of the code in any web application and to run natively in most of the recent browsers without the need for external applications or plugins (e.g. Java RE, Flash, etc.).

The created prototype requires three third party JavaScript libraries: JQuery, D3.js. and Buzz!. More information about each library and technology can be found in section 2.5.

The d3.js library handles the data and provides an abstraction for a particle engine, handling the calculation for the position of each node of the graph. The library also gives the developer the liberty of opting for the use of SVG primitive elements or HTML elements in the creation of the visualization. Both solutions can be combined and have similar results. The main difference between them are the properties that each element offers to the developer (SVG has less properties and selectors than HTML elements, but primitives are easier to place and some allow drawing polygons).

There was no noticeable difference between the speed of rendering when manipulating a large set of SVG elements versus a set of HTML elements. But HTML elements are more customizable than SVG, using CSS stylesheets or by "injecting" more HTML inside existing elements.

jQuery is also used for the manipulation of HTML elements, simple animations, event handling and AJAX callbacks. For audio playback, it is used "Buzz!", a simple Javascript HTML5 audio library that allows the use of Ogg, MP3, WAV and AAC audio formats in webpages (this audio format support changes from browser to browser - for example, Mozilla Firefox (version 24) does not natively support AAC and MP3). This library acts as a simpler API for HTML5 audio elements.

Figure 3.3 displays a visualization prototype, created based in the previous mockups. Here we can observe an audio event (or segment) selected in the center of the image (the

general idea for selecting a sound segment would be through a user's double mouse click action on an audio excerpt representation or by an automatic update while the video is playing, based on events from its timeline - similar to movieClouds "Movie View" [20]) and all nodes connected to this one are similar sound events where the distance and colour of the connection link (edge) is directly related to the euclidean value of similarity between the two connected sounds events. The smaller the link is, the more similar the sound event it is (there are minimum and maximum values defined for the link distance to preserve the readability of the visualization). Also colours are used to express the link distance, were warmer colours express the most similar sounds and cold colours express the most distant, and the shortest links overlap the longer ones (higher z-index value). When the user hovers with the mouse pointer over a node, it increases in size to highlight it as a selection.

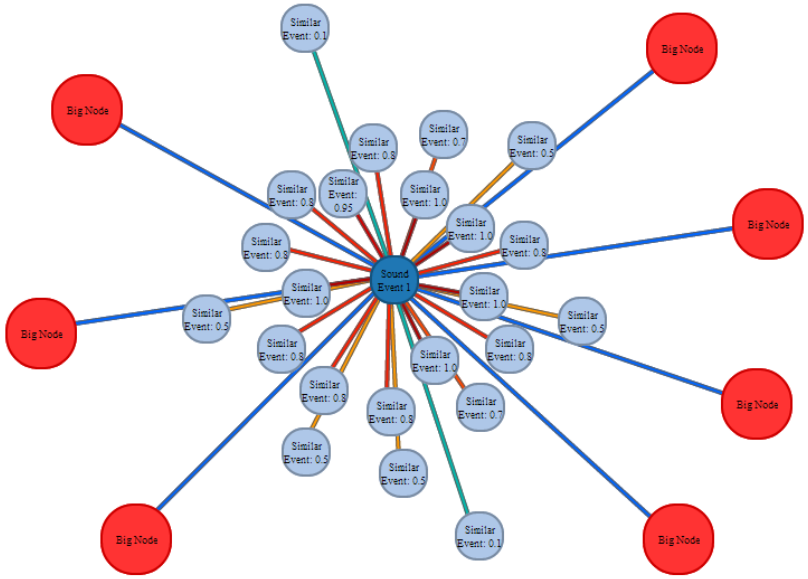


Figure 3.3: Prototype for an interactive Sound Similarity Graph visualization.

The bigger red nodes are meant to be an aggregation of similar audio events in other movies that could be clicked and expanded displaying more nodes. Note that the visualization displayed in Figure 3.3 is created using a fixed test dataset.

For testing the playback of sounds, it was added sounds to the click event of nodes of one of the test datasets available. When the user clicks an audio node, the respective audio segment is played.

To test the prototype's visualization integration with other custom datasets, the prototype allows the insertion of an Last.fm profile user-name with the purpose of extracting the top artists (the most listened artists in number of tracks) from the selected Last.fm user. The used dataset is composed with relations between artists, which are close to audio similarity relations where both represented by euclidean distances between artists and

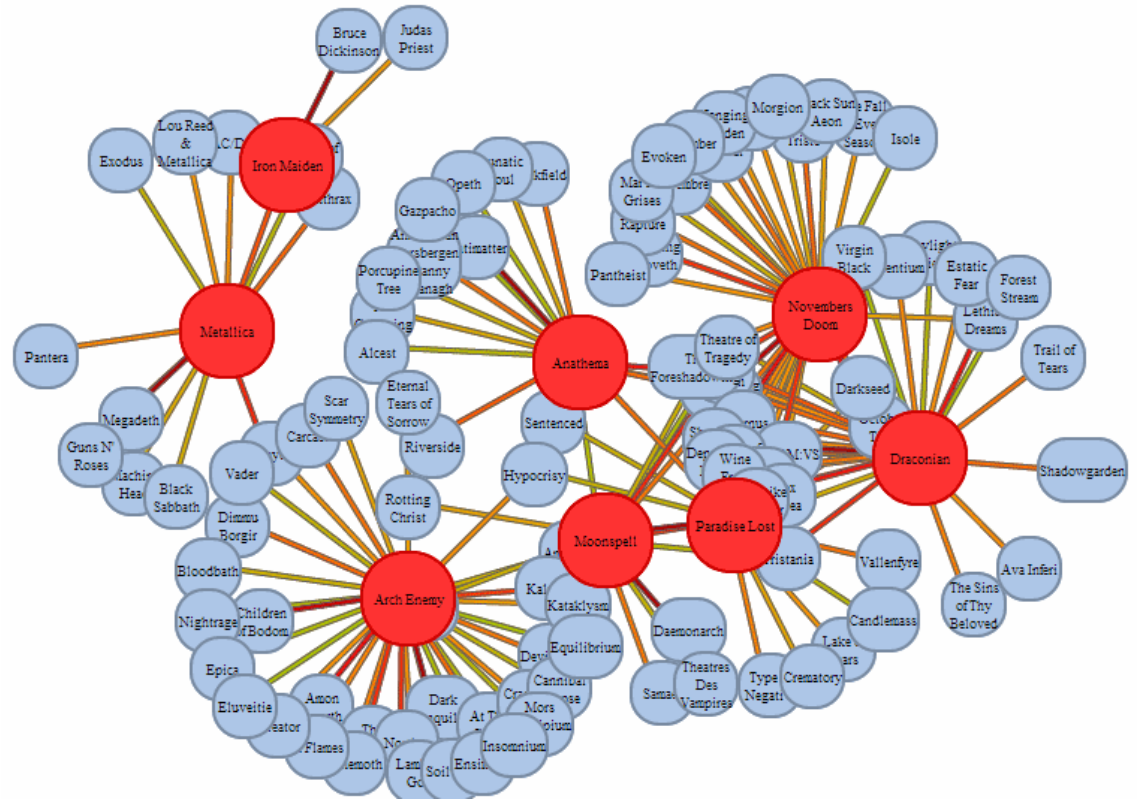


Figure 3.4: The same Sound Similarity Graph visualization using a custom dataset obtained from a Last.fm user profile.

audios (nodes).

The script obtains the top artists from a user profile first, and then obtains the list of similar artists for each top artist to represent their similarity in a graph, as can be observed in the Figure 3.4, using the same characteristics as the graph represented in Figure 3.3, but with the difference that nodes are a representation of music artists and the red nodes are the top artists retrieved from the intended user profile. The data is obtained through the Last.fm API[1] and the similarity value between two artists is provided in each relation returned by the API - a normalized float value between 0 and 1, where 1 is the most similar relation. The obtained result was a mesh of nodes of related artists and links representing similarity relations that can be translated to audio similarity with movies represented as expandable red nodes and audio pieces as blue nodes.

3.1.4.1 Implementation's Software Metrics

Metrics are necessary as a way to measure and compare the effort employed in software development projects and modules and calculate the resources needed for a specific task. Frequently used software metrics are Function points, Code coverage, Balanced scorecard, Number of lines of code, Cyclomatic complexity, etc.

[1] `http://www.last.fm/api`

For a quick measure of the effort employed in the programming tasks during this project, I decided to use the number of Source Lines of Code (SLOC). SLOC is a software metric used to measure the size a computer program through the software source code line counting, used to estimate programming productivity. As advantage, SLOC is an easy task to be achieved and can be automated through simple scripts. However the code used for counting the lines of code should be specific for each language due to their differences, and we have to deal with a lot of issues when using SLOC, such as a lack of cohesion with functionality (a software with fewer lines of code could have more functionalities than other with more lines of code) and is not proportional to the developers experience (experienced developers try to write less but efficient code), the lack of counting standards (What is a line of code? Are comments included? Ternary operators count as one LOC? Is automatically generated code created from the use of GUI included in LOC counting?) and "wrong incentive psychology" (a programmer whose productivity is being measured in lines of code will have an incentive to write unnecessary code).

Language	Files	Blank	Comment	Code
Javascript	3	148	222	1101
HTML	1	18	1	255
SUM:	4	166	223	1356

Table 3.2: SLOC count for the Sound Similarity Representation prototypes.

By using a free command line tool called "cloc" available for Linux based operative systems, I automatized the counting process of lines of code to reach a simple representation of the effort applied in the creation of the prototype and similarity graph library.

The tool supports a wide variety of languages and excludes every line which is blank or only has comments or brackets. But it is not perfect since it does not parse the source files that would solve some ambiguous cases. For the SLOC counting, code from third party libraries were excluded since they do not represent work developed inside this project.

The similarity prototype developed and explained previously in this section includes a total of 1101 lines of JavaScript code (222 lines of comments) and 255 lines of HTML code (see table 3.2). A total 1356 lines of code and 223 comments have been created for the realization of the prototype.

3.2 Temporal Video Visualizations

This section presents some work developed in the context of the MoviClouds with the aim to create and explore new representations of movies, that provide overviews and browsing mechanisms based on the content of movies represented along time.

3.2.1 Motivation

Video is a rich medium, including images, audio and subtitles, which change in time. Visualizations can play an important role in enriching the process of understanding data helping to deal with its complexity and have the potential to make it more aesthetic and entertaining [31]. A time oriented visualization could help to capture, express, understand and effectively navigate movies along time: both the time when they were released or viewed, and the time along which their contents are weaved, in each movie [32];

The visualization of time-oriented data is not an easy task, not only due to differences in the physical properties (time and space) but also due the distinct notions and conceptional models of time and space people develop and the coexistence of several [32].

It is intended to extend MovieClouds and find new ways to represent data about the content of movies along time and their different perspectives with a stronger focus on visual properties to complement the first approach based on tag clouds that represented the content semantics. These representations should be simple but powerful, effective and easy to understand.

3.2.2 Design

The proposed visualizations, intend to provide overviews that represent the whole content of the movies, allowing to see how it changes along time, to compare the different tracks or perspectives and to navigate the contents in order to access the movie at the selected time in a visual and entertaining way. The same visualizations must allow zooming for increased details.

Using concentric wheel shapes, based on the clock metaphor that mimics the way people may perceive passing of time, and Ben Shneirderman's (1996) Mantra "overview first, zoom and filter, then details-on demand" as base concepts, three main visualizations where developed: 1) visualize and explore content tracks or perspectives in a movie, related with image, sound and subtitles with a focus on emotions; 2) to compare movies by selecting content tracks; and 3) to combine contents (e.g. movement and colours) in integrated views when comparing movies.

The first visualizations that were presented in [31] were then extended with dynamic interaction to animate them along time and integrate them with movie and content browsing in MovieClouds. And that is where it crossed the scope of this thesis resulting in a collaboration to enrich MovieClouds even further.

We will now detail the first two visualizations design:

The first visualization (figure 3.5) overviews and allows the browsing and watching of the movies' content through a wheel visualization, named "movie wheel". The movie wheel presents a diverse range of tracks that represent different perspectives of the movies contents. Scenes, motions, colours, emotions in subtitles, felt emotions, mood, audio

events and subtitles are examples of such perspectives or tracks. In the middle there is tag cloud that overviews the selected perspective at any time.

In the wheel, time progresses as in a counter-clock direction, allowing it to synchronize with a linear timeline presented beneath the wheel for more details of the content, and the actual video playing. This straight in figure 3.5 timeline presents the scenes perspective and synchronizes the current position to match the exact same moment represented in the wheel bellow the vertical white line, even when they are dragged by the user when moving the movie timeline temporally forward or backwards (and around). Here we can observe that a concentric wheel shape allows presenting the whole movie with all the perspectives in a shorter space and in a thinner space when compared to the horizontal straight timelines that is better for details about the scene contents.

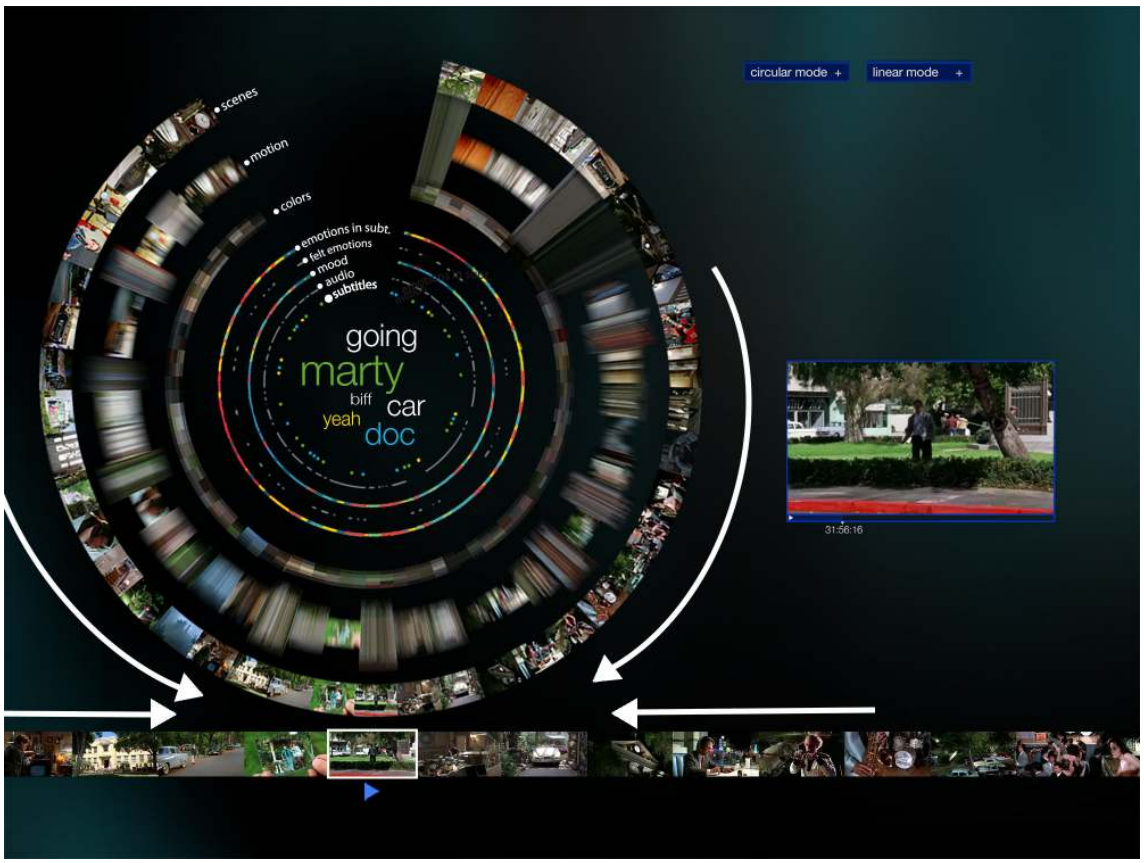


Figure 3.5: Visualizing contents in a movie.

The second visualization (in figure 3.6) allows to relate and compare movies in chosen perspectives or tracks of their content (e.g. scenes images, allowing to get an idea of the main colours and visual rhythm by the number of scenes and image difference). The visualization is similar to the previous one, but wheels and linear now tracks are represent different selected movies only in choosen tracks.

It is possible to select the movie thumbnails in order to watch the movie at that chosen scene and the user may be taken to the previous visualization in order to obtain more

detailed information about each movie. Here the navigation concept and synchronization with the straight timeline is similar to the first visualization. But now the movies will move and synchronize with a white vertical mark that appears in the requested scenes with highest score or frequency related to specific criterion (e.g. most dynamic and coloured scenes of the movie, colour comparison, movies properties, etc.). The scenes related with the same criterion but lower frequency than ones already aligned are highlighted and marked with a red line along the wheel.

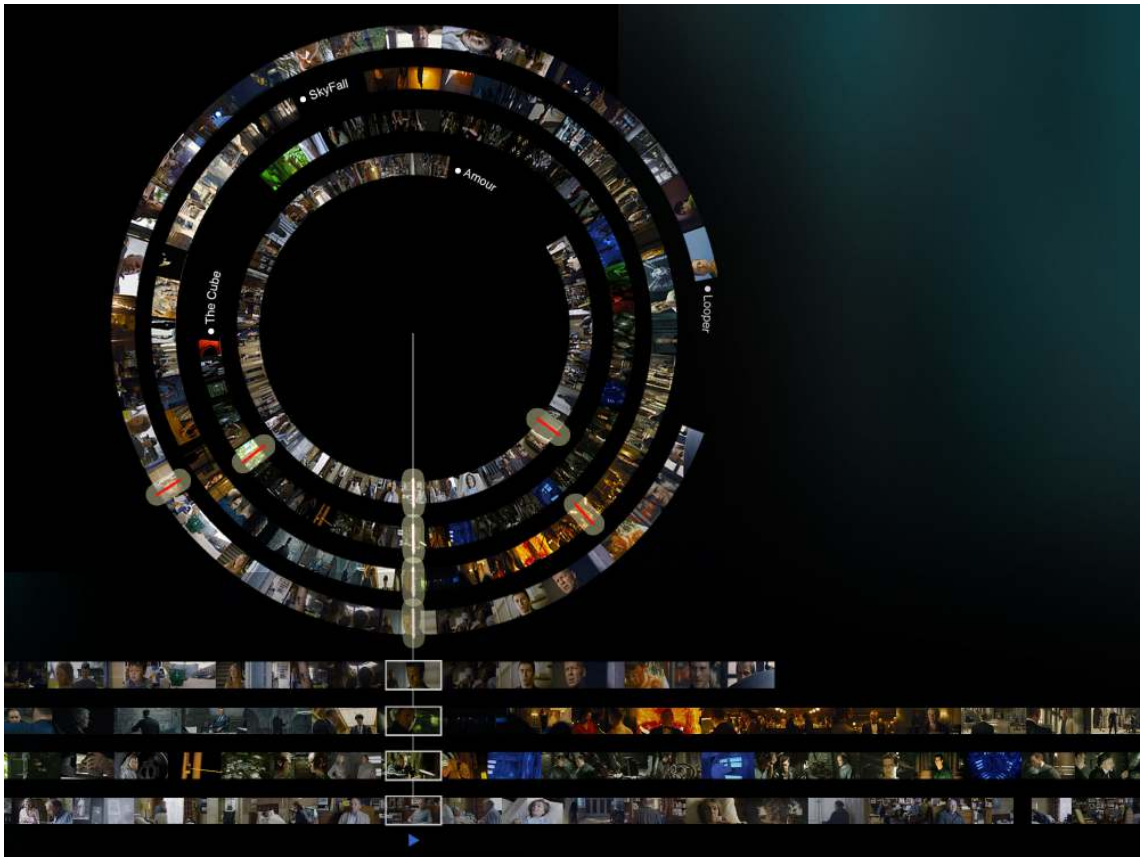


Figure 3.6: Comparing movies by content tracks.

The linear representation can display more information about the movie, but requires horizontal space. In contrast, the circular view can represent the whole movie, or group of movies, in a limited space. However, details in the circular view can be difficult to identify for long movies and massive number of scenes.

3.2.3 Implementation

The prototype has been developed using web technologies and libraries discussed before in sections 2.5 and 3.1.4: HTML5, CSS3 and JavaScript, JavaScript visualization library D3.js to handle the visualization data and manipulate the page content and styles.

The implementation used two third party JavaScript libraries: JQuery and the visualization library D3.js.

The prototype was built around the idea that movies can be decomposed in multiple scenes with heterogeneous duration. The definition of a scene is not going to be addressed here, since this work is focused on the data representation. A simple test dataset was used to simulate six different movies with multiple durations, and distinct number of scenes.

Figure 3.7a displays the representation of a movie, using two different but synchronized views: In the top, a circular (wheel) view from the movie and a linear view of the same movie. Each colour corresponds to a different scene from the movie, and has a time correspondence in both views. The current moment is represented by the vertical red line. Here, colours are not associated to any relevant meaning about each scene, besides the sole purpose of visual separation and highlight.

Notice that the circle perimeter may not be not equal to the width of the linear representation, but this difference does not prevent both visualizations to be synchronized at any moment by moving the same relative space at any time (3.7a-b: in both images the videos are kept in sync at the vertical red line, at the bottom of the wheel).

Figure 3.7: Movie visualization along time.

By synchronizing both views at a specific point, we are free to add multiple movies or tracks to each view, which may change the circle perimeter and present them with different zoom levels - for increased flexibility. In figure 3.8, we observe the same visualization with different tracks. Each track represents a different movie with different scenes. Colours in the image were used to represent the different scenes with no further meaning associated with the colour mapping.

3.2.4 Discussion

The prototype is based on work in progress ideas which are still open being extended.

At the time of writing, new prototypes are being developed using the Processing[2] programming language to evaluate the developed concepts and user experience. These

[2] http://processing.org/

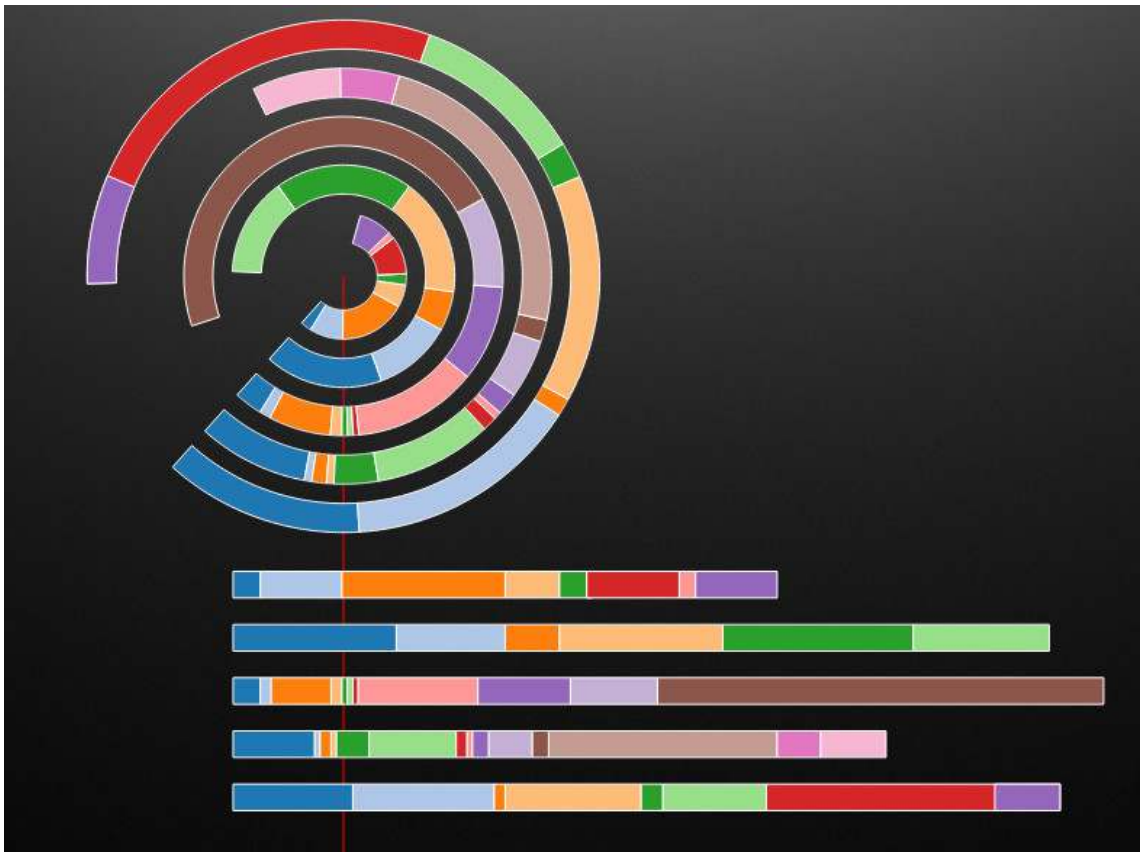


Figure 3.8: Movie visualization along time with multiple movies (track in a cluster).

prototypes can be integrated in MovieClouds using a the JavaScript library Processing.js[3].

First evaluations and users' feedback had very positive results and allowed to identify most strengths and challenging aspects that are being refined along with full development of the designed features.

Future work includes adding exploring more content tracks of information and automotize further content detection, refining and extending current visualization based on the initial goals presented in [31] and observations made through user evaluations, to picture how people perceive the representations and choose new design options to improve and obtain effective, rich and expressive visualizations that can help to provide insights about movies, their contents and their impact on people.

[3] http://processingjs.org/

Chapter 4

Movies Soundstrack Browsing and Labeling in SoundsLike

This chapter presents SoundsLike, a Game With A Purpose that aims to engage users in movies soundtrack labelling to collect data to improve existing content-based sound analysis techniques, and at the same time entertains the user in movie browsing.

4.1 SoundsLike

SoundsLike is a prototype which is integrated directly as a part of MovieClouds for the purpose of classifying and browsing movies' soundtracks. It provides an interface for interactive navigation and labelling of audio excerpts, integrating game elements to induce and support users to contribute for this task, by allowing and simplifying listening to the audio on the context of the movie and presenting similar audios and suggesting labels. It aims to provide overviews or summaries of the audio and indexing mechanisms to access the video moments containing audio events (e.g. gun shots, animal noises, shouting, etc.) and mood to users. To identify such events, it requires statistical models that rely on databases of existing labelled data. However, such databases that would be suitable for this purpose are not available, and the building of new databases would require a massive number of hours of listening and manual classification - the cold start problem referred previously in section 2.4 "Human Computation in Content Classification".

It is desirable to look for other ways to collect such high quantities of information. Therefore, a solution that consists on bringing the human into the processing loop of video and audio through Gamification and a Human Computation approach was explored. This approach is not a novelty, in section 2.4 we presented some previous approaches made before. But it innovates upon these applications both in terms of entertainment aspects and in terms of the definition of the game and the movie browsing in order to stimulate the interest of the user in labelling audio excerpts inside the movies, allowing the collection of data that will help solve the cold start problem for the classification of audio events.

Along this document, the terms "tagging" and "labelling" are introduced to the reader

but they relate to the same concept. The terms "tag" and "label" represent non-hierarchical keywords or terms assigned to a piece of information with the purpose of describing its contents and help finding it through browsing and searching mechanisms.

In this chapter we are going to present the approach behind SoundsLike, including the differentiating aspects to the state of art, requirements, design options and finally the implementation details and architecture.

4.1.1 SoundsLike Requirements and Approach

SoundsLike has the mission of providing a joyful experience to the user and not only focus on the data collecting task. We chose to have a much richer interface compared with the previous applications, which provides for a lot of contextual information about the audio excerpts that users are asked to label.

The application is oriented towards the classification of small pieces of audio of any kind of sound. The audio samples presented to the user are four seconds long, compared with the thirty seconds or more for other games, which gives us the following benefits: shorter samples are easier to classify due to the highly heterogeneous nature of sounds and the unlikeness of having a larger number of sound events mixed up together. If we are able to identify consecutive sound events, we could extend them to longer samples through concatenation of the output of four seconds chunks.

It is also aimed to be played asynchronously over the Internet, not requiring for users to be online at the same time to initiate a game session. This feature increases the complexity required for cheating due to the elimination of most of the communications issues found in previous games (e.g. communication through labels), and allows users to be rewarded while offline when labels proposed by the user are reused by others, thus working as a incentive to return to the game later.

The following functional requirements were taken in account:

- It should be possible for users to label excerpts from movies soundstracks.
- It should be possible to select and visualize more information about a specific audio excerpt;
- It must be possible to visualize, preview and select similar audio excerpts;
- The current selected audio excerpt must be highlighted from the remaining excerpts in the visualization;
- It must be possible to preview an audio or video of the excerpt or in the context of the movie it belongs to, through simple interactions with an excerpt representation;
- It must be possible to select another excerpt present in the visualization to be the new current excerpt.
- It must be interactive, providing a visible feedback for each interaction;
- It should be possible to identify relationships between all components present in the

interface (e.g. clicking over a audio representation would animate or highlight all components that represent that same element in other components);
- The application should be accessible over the internet;
- The data extracted and the gathered information should be accessible and usable by other applications and people without requiring complex libraries and new protocols;
- The application will be integrated in the MovieClouds prototype.

As non-functional requirements the following list was defined:

- The visualization must present the data as fast as possible;
- The application must be easy to use, independently of the user's expertise;
- It must be useful, satisfactory, easy to understand and easy to use;
- It should not overload users with too much information;
- The visualization and animations should be smooth and appealing;

These requirements will inform the design development and evaluation of the SoundsLike prototype.

4.2 Designing SoundsLike

SoundsLike is part of MovieClouds, dedicated to soundtrack interactive browsing and labelling. It integrates gaming elements to induce and support users to contribute to this labelling along with their movie navigation, as a form of a Game With a Purpose (GWAP).

The interactive user interface was designed with the aim of suggesting an opportunity to play and contribute, and to support users in this labelling task, in the access to the videos to classify by allowing and simplifying listening to the audio in a context of a movie, by presenting similar audios and suggesting tags. For this purpose, the visualization explored in section 3.1 is reused and extended.

The following sub sections will present the main design options for Soundslike, by labelling sound excerpts in the context of a movie navigation from MovieClouds. For more information about navigation inside MovieClouds, see section 2.6.2 - Movie Clouds.

This explanation starts in a scenario where a user selected the Back to the Future movie in MovieClouds, ending up in Movie Views in Figure 4.1a) where he is able to play the movie, and at the same time visualize five timelines for the content tracks showed bellow the movie and a selected overview tag cloud of the content presented on the right (audio events in the example), synchronized with the movie that is playing and thus with the video timeline.

In a situation where the user is registered and authenticated, a small Heads Up Display (HUD) is always visible in the top right corner of the screen with the name, accumulated

points and rank level (based on points: rookie in the example) to the left of the SoundsLike star logo, reminding the user to access it.

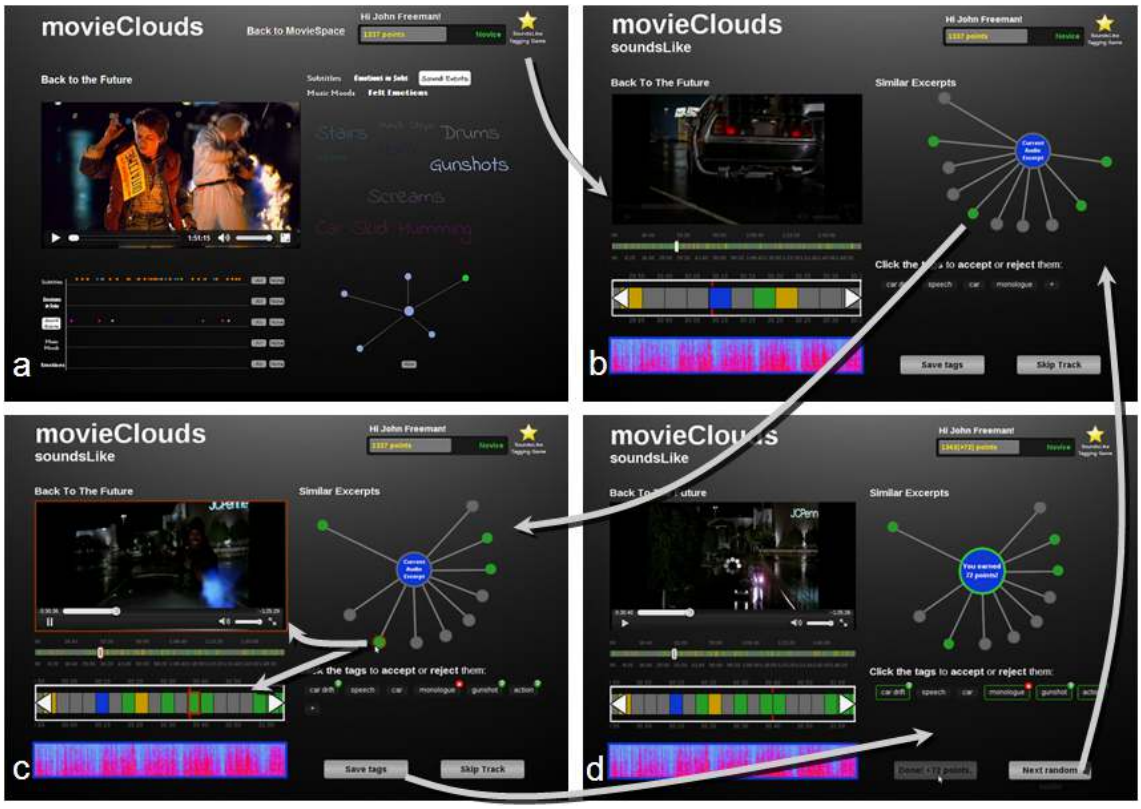


Figure 4.1: SoundsLike interaction: a) audio events track in MovieClouds Movie View; b) selection (click) of a neighbour excerpt; c) to be played as video, providing more context to identify the audio (excerpt highlighted with brownish red frame on the graph, timelines and video); d) saving labels, winning from labelling and choosing to play again and tag once more.

4.2.1 Playing Soundslike

After pressing the SoundsLike star logo, the user is immediately challenged to label audio excerpts from the current movie. An audio excerpt is highlighted in three audio timelines with different zoom levels bellow the video, and represented, to the right, in the center of a force oriented graph displaying similar excerpts, with a field for selection of suggested or insertion of new textual labels bellow, to classify the current audio excerpt, and hopefully earn more points.

At this point, users can preview the video and audio for the current audio excerpt and notice consecutive excerpts at the timeline or similar excerpts at the similarity graph. Then they will be able to decide to label the current excerpt by accepting labels from the list of suggestions or add new custom labels, or skip the current excerpt and select another excerpt for labelling.

By allowing to display the current excerpt and surrounding neighbours, inspect details of audio signals, and to navigate and listen to entire audio excerpts and possibly watch them in the context of the movie, SoundsLike was designed to support the identification of the current audio excerpts to be labelled.

4.2.2 Movie Soundtrack Timelines

Right bellow the video, three timelines are presented (Figure 4.1:b-d, close up available in figure 4.2). The first timeline on the top, also known as Soundtrack Timeline, represents a summary of the entire soundtrack or video timeline for the current movie. In this view, users can visualize what areas of the movie they have already classified, with the current position highlighted in the context of the whole movie.

The second timeline presents a Zoomed-in Timeline view with the position and level of detail selected by the user, by dragging a white marker on the Soundtrack Timeline (Figure 4.2:b-c). This view presents a selection of the information in the first timeline in more detail. The selection can also be moved by clicking at the arrows visible at each side of the zoomed-in timeline. Likewise, the marker can be expanded and shrunken to include more or less audio excerpts in the selection.

The third timeline represents a close-up over a chosen excerpt as an audio spectrogram. Spectrograms are a visual representation of the spectrum for frequencies in a sound and can be used for example to identify music instruments, spoken words phonetically, or analyse the various calls of animals [24].

The representations here are designed to be similar to the timelines already used in MovieClouds for content tracks. Here, audio excerpts are segments from the entire soundtrack and are represented as separated rectangles arranged sequentially.

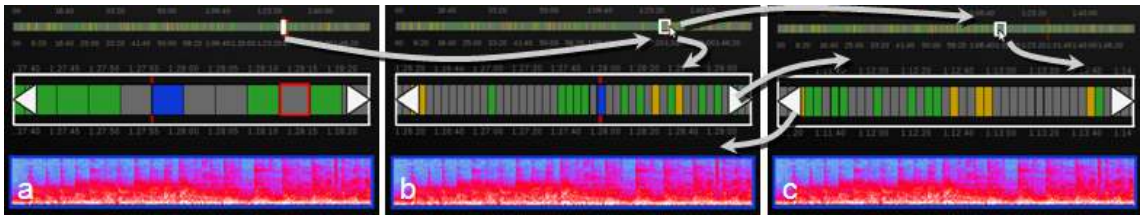


Figure 4.2: SoundsLike Timelines: a) Soundtrack Timeline, Zoomed-in Timeline and current audio spectrogram; b) Zooming out/in the zoom timeline by dragging the marker open in the Soundtrack Timeline; c) Dragging the marker right to make the zoom timeline move ahead in time. The arrows do the same. The markers are always synchronized in all the timelines.

Relationships between elements are represented and reinforced through colours. The current audio excerpt is highlighted in blue, while grey represents excerpts not yet classified by the user, green refers to excerpts previously classified by the user, and yellow refers to excerpts which were visited and skipped. A small tooltip reinforces the asso-

ciation by describing the state for each excerpt (classified, unclassified, skipped, current audio excerpt) when hovering with the mouse pointer.

All timelines are synchronized between themselves and the video area. A vertical red line marks the current temporal position in every level of the timeline, synchronized along time. Playing, stopping and changing the position of the video will change immediately the red line position in each timeline. The relation between each of the timelines is reinforced by colour matching of selections and frames: the white colour used in the selection marker in the Soundtrack Timeline matches the colour of the Zoomed-in Timeline, and the colour for the current selected audio excerpt (blue for the current audio), matches the colour of the spectrogram timeline frame.

4.2.3 Audio Similarity Graph

A timeline represents the audio excerpts of a movie organized by time. Users can listen to consecutive audio excerpts. To improve the classification, we integrated a view to display similar audio excerpts related to the current selected excerpt.

We named this view as "Audio Similarity Graph", which integrates the work previously described in section 3.1 for MovieClouds. The view is composed by a force-directed graph (figure 4.1:b-d and figure 4.3:a-c), representing the similarity relations to most similar audio excerpts in the movie. The graph is based on physical particles, where the nodes represent audio excerpts and repel each other, tending to spread the graph open to show all the nodes. The similarity distance represents the difference in audio content between two audio pieces and it is translated to the graph metaphor as the screen distance between connected nodes. The graph metaphor is explained in more detail in sections 3.1.3 and 3.1.4.

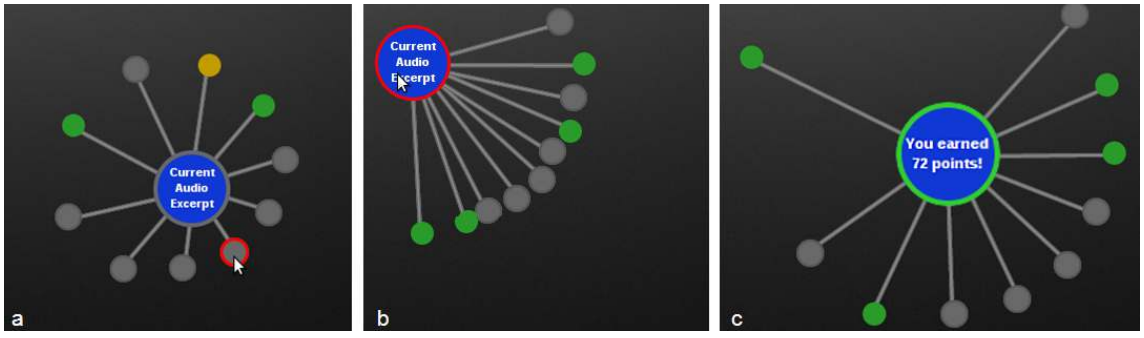


Figure 4.3: SoundsLike Audio Similarity Graph: a) listening to a neighbour (on over); b) dragging the graph around and comparing distances; c) earning points from saving tags to label the audio excerpt (current audio frame becomes green, meaning it was already labelled).

The graph uses the same colour mapping as the timelines, adding a coloured frame (grey, green or yellow) to the current excerpt to reinforce if it was already classified or

skipped. Likewise, hovering with the mouse pointer over a node will playback the associated audio for a quick preview, while clicking it will play the associated video. It is possible to drag the graph around by holding the nodes to compare distance more easily (figure 4.3b).

The user can also double click over a node to select it as the current audio excerpt to be classified. This is particularly useful, if users identify this audio better than the one they were trying to identify, and had not classified it before, allowing to earn points faster, and to select similar audio excerpts after they have finished the current classification.

It is possible to expand the similarity relations to other movies as future work, both based on similarity values or information gathered from labelling. But the current approach detailed in section 3.1 used for the calculation of the excerpts distance in terms of content does not scale well in an environment composed by multiple movies - it would require huge amounts of audio processing and information by comparing the current excerpt with every other existing audio excerpt.

4.2.4 Listening to Contextualized Audio in Synchronized Views

Every view integrated in SoundsLike presents a synchronization of behaviour across the whole interface. Besides the synchronization of the timelines explored in section 4.2.2, SoundsLike adopts the same and updated colours for the same representation in multiple views. For mouse-over action, a bright red coloured frame is displayed in every view to temporarily highlight the locations of excerpts everywhere while the current audio plays. In addition, when an audio excerpt is selected to play in the movie (single click in a excerpt representation in all views), a brown frame is set for the excerpt in the graph node and timelines and this time also around the video area (figure 4.1c).

4.2.5 Labelling Scoring and Moving On

Soundtrack labelling is the most important requirement of SoundsLike. Here, labels are non-hierarchical textual terms with the purpose of describing the acoustic content of an audio excerpt.

Bellow the similarity graph, the labelling area can be found, displayed in figure 4.4 and visible in figure 4.1:b-d. To label the current selected audio excerpt, the user choose one or more textual labels to describe in the labelling area. Labels may be selected from a list of suggested labels, or introduced by the user (selecting the “+” option - figure 4.4b) - which substitutes temporarily the button with an input text-box (figure 4.4a) as a sequence of comma separated labels.

Choosing a label, either to select, reject or ignore, is done through clicks until the desired option is on (green $\surd$ marker to accept, red $\times$ marker to reject, grey to ignore). In addition, labels that were introduced by the user have an additional frame in the same

colour of the marker (green, red). Hovering with the mouse pointer over a label will also display a small tooltip indicating its state and respective short description.

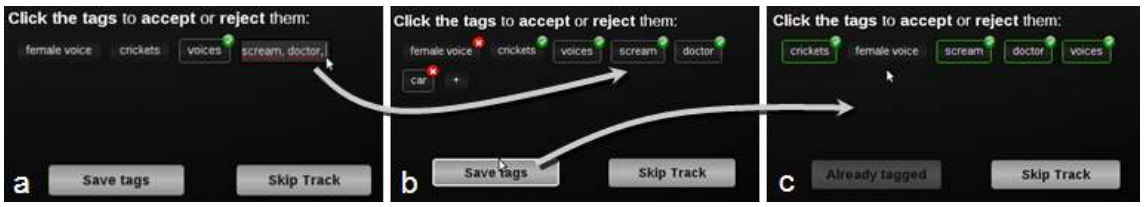


Figure 4.4: Labelling the audio excerpt: a) two labels suggested, one candidate for insertion (the one with the grey frame), and introducing new ones in a comma separated list; b) accepting ($\surd$) and rejecting ($\times$) tags; c) labelled audio with three tags introduced by the user, a suggested tag accepted and another suggestion ignored.

At any time, the user can skip an excerpt and choose another one, or simply leave the game. To save the choices, the user must press the save button (figure 4.4:b-c), and choices are submitted to the server, changing the colour of the current audio excerpt to green in the similarity graph and timelines, and displaying the earned score at the central node of the graph which is also enlarged (figure 4.4d) to highlight the state. Now, the user may use the Next Sound button to randomly navigate to another segment in the same movie, or may choose a specific excerpt from the similarity graph or from the timeline. Excerpts already labelled (green) cannot be labelled again (the rational behind this restriction will be explained in the next section).

4.2.6 Gaming Elements

This section presents the main design ideas behind SoundsLike to induce and support users to contribute to soundtrack classification of movies. Here a gamification approach is used to incite users to classify audio, finding ways to engage the rewards mechanisms for doing a good labelling, in the same ways that successful video games do, and compel users to come back again.

Gamification typically involves some kind of reward to the users. In SoundsLike, points are used as behaviour reinforcer to incite users to keep contributing into the labelling process, but as it was analysed in section 2.4.4 - Gamification and Human Computation, giving blindly points is simply a bad choice because users will simply get bored and leave after performing the same action over and over to receive purposeless rewards. To mitigate this effect, the reward system is based on how well the user is doing labelling audio pieces. Users should be rewarded based on their skills and not simply by pointless amounts of work: Quality is preferable over quantity.

The reward system works with points and ranks. Points are attributed based on agreement: when their labels correspond to an existing consensus by partial or entire textual matching, and when a label previously introduced gets confirmed by others. Points are

updated in the HUD located in the top-right area of the screen, increasing a progress bar which is related to the missing points required to obtain the next rank.

Lets start by a simple scenario to explain the reward system: User "Alice" reaches an unclassified audio excerpt. Note that the system does not make any difference between audio excerpts which have been classified by other users or not. She classifies the audio with the following labels: "goats", "bells", "farm" and "dog". We could reward Alice with points just by participating to motivate the user to continue her work, however the number of points rewarded by plain participation must be limited by adding some sort of restriction (a limit by excerpt and a daily limit), otherwise, Alice would feel compelled to keep adding tags pointlessly - quantity over quality. A few minutes later, "Bob" reaches the same audio excerpt and labels it with the following tags: "goats", "animals" and "cows". In this case, bob agrees with Alice, on "goats" without knowing about this fact, and he is rewarded with points - lets assume 1 point for each match. At the same time, Alice gets its tag confirmed and receive also 1 point. Later, "Charlie" classifies the same excerpt with "animals, "goats" and "noises", matching "animals" with Bob's labels, and "goats" with Alice and Bob. Charlie is rewarded with 3 points, while Bob receives 2 points for getting two labels confirmed, and Alice one point.

users by the quality of the labels through a crowd-sourcing approach. Points received by labelling confirmation are given asynchronously, not requiring the user to be active at that specific time.

Some aspects are not final and require a detailed evaluation to investigate possible advantages and problems. Games usually stop being fun when people find them too easy, the rewards are not fair compared to the required time and skills for the task, or users find ways to get advantages over other players, by cheating or "bug abusing".

In the cheating scenario, users may try to add a countless number of labels to each audio segment in an attempt to raise the probability of receiving points. This problem can be mitigated by restricting the maximum number of labels a user can suggest per excerpt, forcing users to describe audio with more precise labels. Another measure aiming to keep labels as simple as possible, would be restricting tags to a maximum number of characters, preventing users from writing huge declarative labels which are unlikely to match with other players and hard to analyse through automatic mechanisms (this restriction is already implemented and defined in the database model).

Another cheating scenario would be through "word poisoning": a significantly huge group of users may try to write the same group of words in every audio segment to get an advantage over other users (e.g. they may introduce "aaaa" in every excerpt). This problem is not trivial to mitigate, but was observed and analysed in the ESP game [64], named by the author as "massive agreement strategy" and can be detected by monitoring changes in the the average time in which players are agreeing on labels. The dictionary restriction would help in this case by adding some restrictions that prevent the introduction

of meaningless random words.

Finally, by restricting words in labels to a dictionary would prevent most erroneous labels that would be introduced due to unfortunate misspells, the use of another idiom, or random gibberish introduced on purpose.

A summary of key points, discussed in this section, until now:

- Points are rewarded directly by matching inserted labels with existing consensus;
- Points are rewarded asynchronously by getting labels confirmed by other users;
- Points could be given by participation in unclassified audio excerpts, but we must be careful and limit them;
- The accumulated points by the user are displayed in a HUD on the top-right area of the screen;
- The number of labels a user is able to introduce per excerpt must be limited to prevent abuse and improve label quality;
- Labels must be limited to a maximum number of characters to maintain them simple, more likely to be matched and appealing to automatic data analysis algorithms;
- Labels must also be limited to dictionary words to prevent erroneous and useless labels;
- Massive agreement strategy can be detected by monitoring changes in the the average time in which players are agreeing on labels;

Both “game is too easy/hard” and “cheating” scenarios can be related directly to the Csíkszentmihályi Flow Theory [9] (section 2.4.3), where a game that is considered too easy, i.e. tasks are no longer complex to the current user’s skill causing a feel of relaxation or boredom, or rewards do not match the difficulty of the task (or the task is simply too complex for the user’s skill) create a feeling of worry or anxiety. The cheating scenario also relates to Flow Theory by using unexpected situations (bugs or unforeseeable features) to gain advantage over others by greatly reducing the task’s complexity (situations where a user would receive a greater rewards than the usual is also reducible to a complexity decrement).

To engage users continuously into the task of audio labelling, approaches are required to keep them experiencing Flow while contributing. For this purpose, a refinement and evaluation of scoring and rewarding mechanisms are needed to observe if points are rewarded fairly to the effort spent by users and introduce gradually them to new social and gaming elements (single and multi-player challenges, charts, ranks, avatars, medals, achievements) as their skill increases, giving an increased sense of purpose to received points. Users are not going to engage to simply receive points!... or at least most of them will not after a certain time accumulating points pointlessly.

A further analysis is required of how to maintain users engaged in cases where they are not immediately rewarded and must wait for others to receive points. To compel users to comeback, we could inform them periodically (maybe daily or weekly), through email

or other communication channels, about the points obtained through confirmation since their last visit, and by inviting them to open the web application to receive them. We could require them to classify some audio segments daily (or a greater time period) to receive the confirmation points from the same day - this would be similar to Pay To Click services where users can refer new users (some may allow to buy referred users) and receive a small percentage of each click made by them as bonus, but users must visualize a daily quantity of advertisements to receive the next day bonus from their referred users - but and again, we should analyse how this could affect motivation in short and long term. Improved animations would take a important role inciting users by displaying a visual and rewarding feedback. Ranks (and even points) could also be used to unlock other application features or reducing restrictions. However, not everyone values the same things! It is also necessary to pay attention to the cultural contexts and values [34], involving balanced intrinsic and extrinsic motivations and rewards.

SoundsLike project focused mainly on the user interaction and visualization and will benefit from future studies and the addition of features to improve user engagement. Points could be used as an exchange coin to obtain products and services from third party retailers (cinema tickets discount, movie and music streaming service discounts, t-shirts, gadgets, etc.) or by exchanging for new features in the website (new tools, removing restrictions, extra social features, points bonus). The introduction of leader-boards and the integration with social networks such as Facebook would also open to competition between users and friends that could be engaging.

4.3 VIRUS System Architecture and Implementation

SoundsLike and MovieClouds are frontend Web applications for the VIRUS project.

One of the requirements proposed in this project pointed that the gathered data should be easily accessible by other people and applications. Hence, my approach to the VIRUS system architecture consisted in the separation of the application in a three tier architecture, represented in figure 4.5, which allows the decoupling of components from the upper layers: The Presentation Tier, the Logic Tier and the Data Tier.

End user applications will fall under the Presentation Tier and will communicate with the Logic Tier through a pre/defined interface. Developers who desire to use services and data from the VIRUS database just need to interact with the logic tier using the most suitable intermediate language and interface. Any changes in the lower tiers will not affect the applications logic as long the interface between the tiers is not changed (features cannot be removed from the API).

The Presentation Tier is composed by any kind of application that will require data stored in the VIRUS database. It includes web applications, graphical desktop applications and server applications with or without graphical interface (e.g. data aggregators

and indexers).

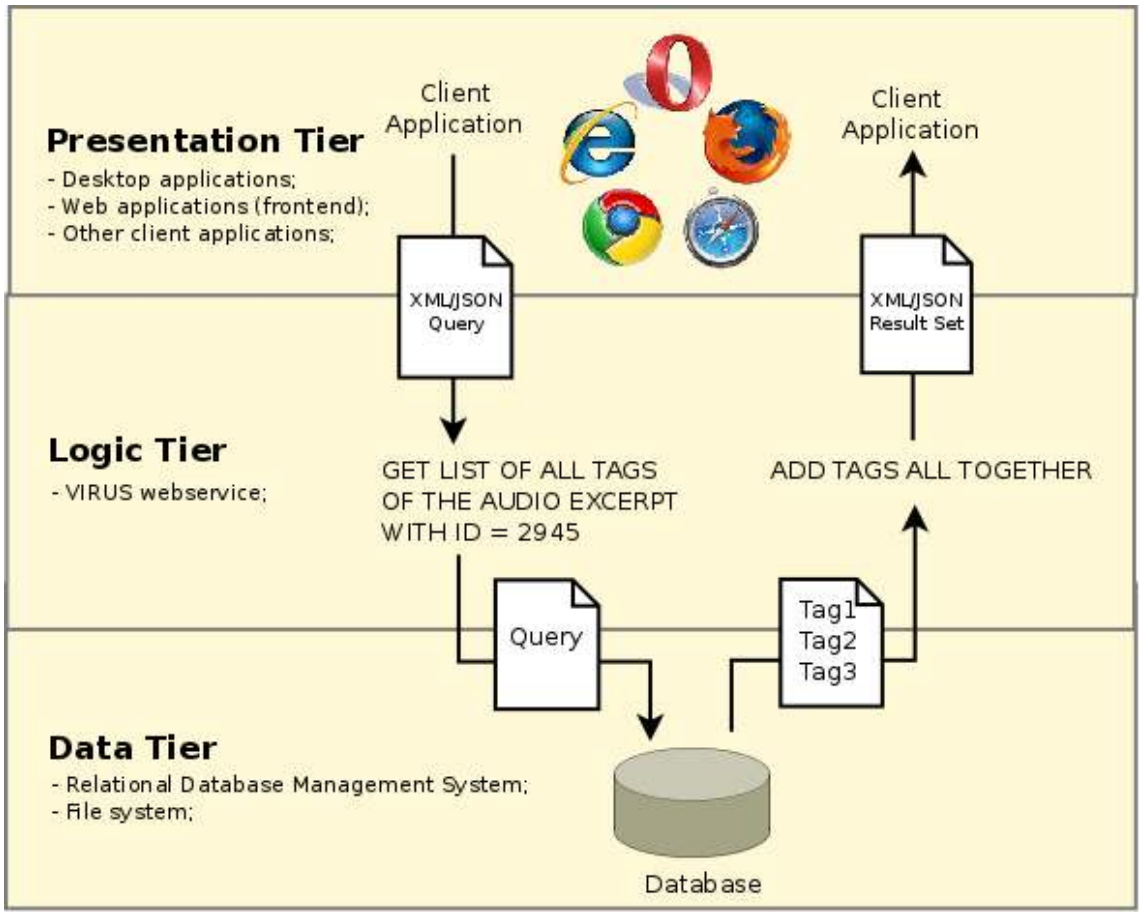


Figure 4.5: VIRUS system tier architecture.

The Logic Tier components will handle requests coming from the Presentation tier and will validate them before returning a response. Client authentication will also happen here to obtain access to certain resources and features.

All operations here should use the minimum amount of resources and must not affect other user sessions. Components from the upper layer must receive responses to their requests as soon as possible. Operations that demand for large amounts of time to complete (e.g. extraction of audio features) should be scheduled to be processed asynchronously and queried later.

If needed, a query will be sent to the lower layer to obtain information from the database or resources from the disk. An application programming interface (API) should be provided to the Presentation Layer for the development of applications apart from the used programming languages.

The Data Tier represents the components who will handle and store information in a persistent storage. If needed, components here will also handle data replication and load balancing to ensure the availability of the system. The data tier will communicate with the Logic tier to retrieve and store information from the storage. In this tier, the storage

system can be fully replaced by others as long the interface provided to the upper layer is maintained. It is possible to use multiple storage systems for multiple purposes (e.g database management system for storage of big quantities of structured information and file system for storing multimedia resources)

I'm now going to demonstrate the implementation chosen for each tier starting by the bottom layer.

4.3.1 Data Tier: VIRUS Database

The VIRUS database purpose is to store information produced by SoundsLike and others applications integrated within the system. It will store details about the available movies and respective resource files (video, images, audio and subtitles) in the file system, user accounts, audio excerpts, audio similarity values (computed by comparing values from the extraction of sounds features and all data gathered inside of SoundsLike), and audio labels.

MovieClouds used simple text files for storage. This approach does not scale and may suffer from performance issues and inconsistencies when faced with multiple operations and data accesses happening at the same time. A better solution is required in which operations should follow ACID (Atomicity, Consistency, Isolation, Durability) properties, the system should handle multiple operations at the same time and support data replication (if necessary). The database management system chosen for this effect was MySQL.

Before advancing further into this choice for the storage system, it is worthy to mention the existence of structured storage alternatives to a relational database management system (RDMS). Such alternatives includes NoSQL databases and ontologies (e.g. Web Ontology Language). However the advantages and disadvantages for each system were not evaluated.

MySQL is considered the world's most widely adopted open-source relational database management system [71], which uses SQL (Structured Query Language) as interface for managing stored data. The main reasons behind the choice of MySQL were: the licensing price - MYSQL is free software; the popularity - most internet hosting environments provide MySQL hosting without requiring an extra fee; cross platform - MySQL can be deployed in Windows, MacOS, Linux, FreeBSD and Solaris operative systems; and the experience I had in using MySQL before.

Figure 4.6 shows a UML Entity–Relationship model, created using an utility software called MySQL Workbench[1], that represents the current structure of the VIRUS database. This database stores information about videos, users, audio excerpts from videos (SoundSegment), similarity relationships (SoundSimilarity), and labels (SoundTag) given by users to audio excerpts. Here we define videos as composed by multiple sound segments. Each audio segment has a start and an end time, in milliseconds. Each segment may have

[1] `http://www.mysql.com/products/workbench/`

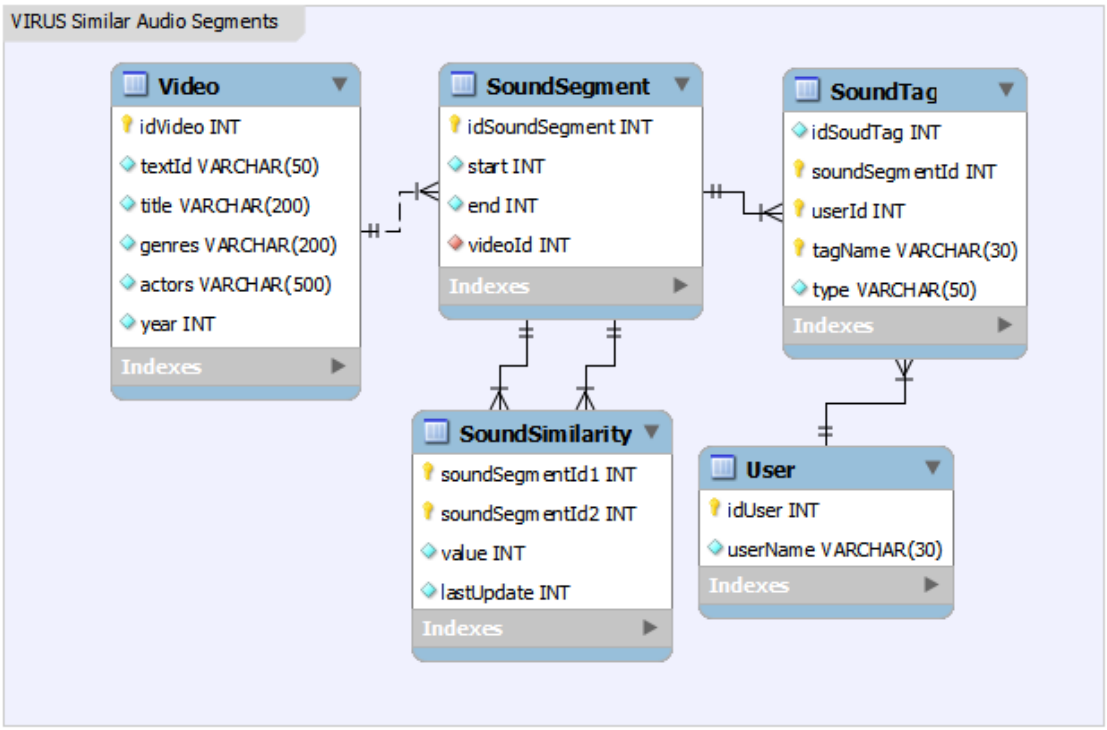


Figure 4.6: UML Entity–relationship model for the similar audio segments database and tagging system.

a similarity relationship to another segment, and may also have multiple labels (or tags) given by multiple users. However users cannot associate the same tag twice to the same audio segment.

This database structure is enough for holding the data collected from SoundsLike prototype, but it is limited to audio and video pieces, and therefore not extensible to other multimedia types.

For future work, a new and improved database structure would be required to hold any kind of multimedia documents and relationships. In figure 4.7, it is proposed a second version of the database that would be able to store more complex information and most relationships identified in multimedia documents.

In this version, multimedia documents (MediaContent, i.e. music track, movie) can be separated in a diverse range of multimedia excerpts or pieces (MediaPiece). Each piece may possess multiple tags (MediaPieceTag), given by Users. Each tag may have a type (e.g. emotions, actions, animals, instruments) and a confidence level (positive or negative, may be used to differentiate tags obtained from automatic information retrieval mechanisms from manual classification).

Also media content may be included in different collections and belong to a group of authors (MediaAutorGroup, i.e. music group, movie cast). User may be artists who participate in multiple movies or music bands (group of authors).

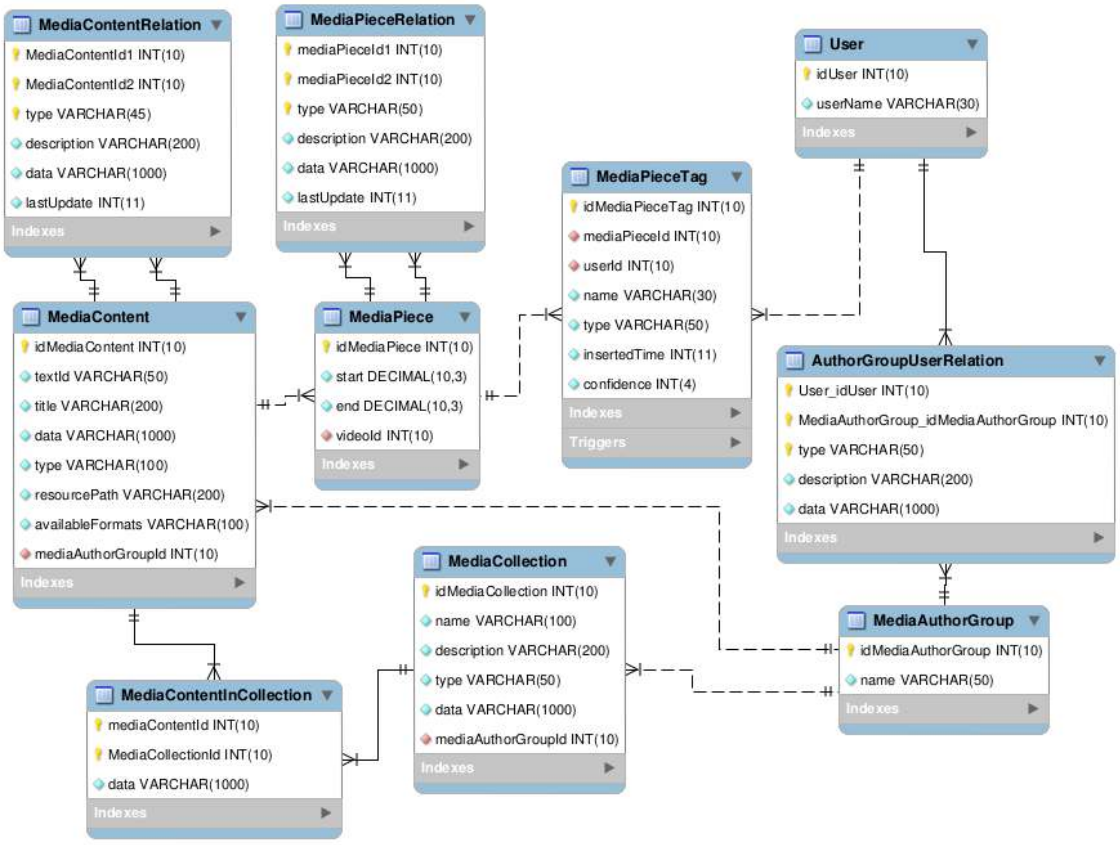


Figure 4.7: Proposal for second version of the UML Entity–relationship model for the similar audio segments database and tagging system.

In this database, relationships between multimedia contents, multimedia pieces, and author groups are possible, and share the same fields: type (used to select different kind of relationships, however you cannot have the same type of relationship between the same two elements), short textual description, and the data. The "data" field in a relationship is designed to hold a variable sized string that corresponds to a simple JSON data array with properties and values (e.g. store the similarity value between two pieces, "X was bass guitar player in group Y from time A to B"), enabling a wide variety of relationships compared to the previous database model.

4.3.2 Logic Tier: VIRUS Webservice

The logic tier is composed by a remote procedure call (RPC) mechanism, in this case a RESTfull Web Service. Remote procedure call is a communication process that allows a program to generate an execution of a method or procedure in another machine, in the same or other network. The specification for this iteration must be independent of the location where the method will run.

The purpose behind this system is to incite people to build and test applications that

use the data provided by the project, because these mechanisms remove most of the communication difficulties between independent components in different heterogeneous environments by adhering to wide known simple communication technologies, semantics and protocols, and not needing extra mechanisms above the RPC layer. By other words, it will allow applications to integrate data obtained through SoundsLike without the need to implement complex communication mechanisms by using technologies implemented by libraries available across most systems. Likewise, the Web Service comes as an alternative for the distribution of audio classification datasets. These datasets would be useful for the construction of statistical models of data to be employed in Information Retrieval mechanisms. Instead of exporting huge datasets, these mechanisms will be able to filter and select relevant and updated data, avoiding the overhead of synchronization of and search in large datasets.

Due to the fact that I was building web applications, I opted for the creation of a REST-Full web-service and the respective client side API for web applications, using JavaScript.

According to the W3C, web service is defined as "a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artefacts. A Webservice supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols". A web service supports direct interaction with other software agents by using XML based message through Internet protocols. A web-service is a component that can be integrated in complex distributed applications [1].

The web-service is implemented using the PHP language and communicates to the data tier using SQL language. The current implementation uses PDO (Php Data Object) which supports a wide variety of database driver (including MySQL, Oracle, PostgreSQL, SQLite, Microsoft SQL, etc.) as a data-access abstraction layer. By using an abstraction layer, it is possible to decouple the web service from the data layer. However, It is important to note that PDO supports SQL based solutions but there is no official driver support for NoSQL solutions.

Appendix C presents a short documentation for the VIRUS Web-service to help developers to integrate their applications with the web-service.

4.3.3 Presentation Tier: SoundsLike Front-end

The SoundslLike Web application provides interactive tools to help the user in the task of soundtrack classification and works as front-end application for the VIRUS Web-service.

Soundslike is designed and implemented to be executed in a common Web browser. Details about design options were discussed in sections 4.2: "Designing SoundsLike" and 4.2.6: "Gaming Elements".

The front-end is implemented using a variety range of web technologies and libraries. It uses HTML5, CSS3 and Javascript. HTML5 is used to define the content of the web

page, CSS 3 for styles and some animations, and Javascript to control and manipulate the entire application content, process different kinds of data, for defining visualizations and animations, and to communicate with the Web Service. Note the fact these three technologies are deeply correlated and integrated.

The created front-end prototype required some third party Javascript libraries: Modernizr, JQuery, D3.js. and Buzz!. Modernizr is used to check the availability of new HTML5 features in the client browser. JQuery is used for page content manipulation, animations and handling asynchronous communications with the server. D3.js a framework to manipulate documents based on data, used for the creation of timelines and similarity graph visualizations. More information about each library and technology can be found in section 2.5 - Web Technologies and Libraries.

4.3.4 Implementation's Software Metrics

Here it is presented SLOC metrics to quickly measure effort employed implementing the SoundsLike prototype. The chosen metric was discussed before, in section 3.1.4.1.

The value was calculated using the tool "cloc". Please note the source files of third party libraries were not included in the calculation. In table 4.1, the VIRUS webservice includes a total 4475 lines of code and 1181 lines of comments in PHP.

Language	Files	Blank	Comment	Code
PHP	29	784	1181	4475

Table 4.1: SLOC count for the VIRUS webservice.

Table 4.2 displays the SLOC count for the SoundsLike frontend source files. The prototype includes a total of 3144 lines of JavaScript code (626 lines of comments), 913 lines of CSS and 210 lines of HTML code. A total sum of 4267 lines of code and 842 comments was identified by the counter tool.

Language	Files	Blank	Comment	Code
Javascript	8	317	626	3144
CSS	3	134	183	913
HTML	1	25	36	210
SUM:	12	476	842	4267

Table 4.2: SLOC count for the Sound Similarity Representation prototype and library

4.4 Discussion

In this section we described SoundsLike, a new game with a purpose whose objective is to collect labels that characterize short audio excerpts taken from movies. The interface is integrated in MovieClouds and is designed to entertain the user while pursuing the data collection task.

The proposed interface innovates with respect to previous GWAPs with similar objectives by providing a rich context both in terms of temporal aspects through three timelines with different time scales, and in terms of similarities between items by displaying an dynamic force-directed graph where the neighbourhood of the current item is represented. This Graph allows the exploration of nearby items in terms of similarity but it is also used to navigate through the set of audio samples while always keeping track of the corresponding scene in video document.

It uses a gamification approach to induce users to contribute to soundtrack classification of movies, rewarding them with points when their labels correspond to an existing consensus by partial or entire textual matching, and when a label previously introduced gets confirmed by others. Some cheating scenarios are also presented together with measures to mitigate them, and conclude that a reward mechanism solely based on points are not able to keep users in Flow state, and therefore cannot engage users in the repetitive task of audio labelling for long periods of time. A refinement and evaluation of scoring and reward mechanisms is required to observe their effectiveness and introduce new game and social element to increase engagement.

The next chapter will access SoundsLike interface and its features to detected usability problems and obtain user suggestions.

Chapter 5

Assessing SoundsLike

An evaluation with users has been performed to assess SoundsLike interface, its features, and their perceived usefulness, satisfaction and ease of use. In the process, detected usability problems and user suggestions could inform us about future improvements in the application or new approaches to be explored.

The evaluation aimed at finding interaction and usability problems and perceive tendencies in users acceptance and preferences.

5.1 Method and Participants

The evaluations were conducted as interviews, starting off by presenting a paper sheet to the user (see page 2 of Appendix D), containing a simple questionnaire to obtain simple demographic information about the participant, and their characterization as target audience for statistical purposes, and a summary of the application.

The demographic questionnaire focused on age, gender, computer experience, movie visualization habits online, habits of browsing movie information websites (e.g. using imdb.com), if users had previous contact with MovieClouds. The summary included a brief description of the application's purpose, the starting scenario for the evaluation, and a simple diagram identifying the name of the interface's main components (video, timelines, graph and labels) without any further details to prevent tampering with the results of usability tests.

A task-oriented approach was conducted to guide the user interactions with the different features, while the interviewers were observing and taking notes of every relevant reaction or commentary provided by the participant. Every task involved a set of predetermined features and its evaluation was focused on the user experience, emphasising perceived Usefulness, Satisfaction and Ease of Use (based on the USE questionnaire [2] and using a 1-5 Lickert scale [38]).

At the end of each task, the interviewers asked for suggestions and USE evaluation of every relevant feature. In the end, users were asked to rate the application globally, refer

to the aspects or features they liked the most and the least, and classify it with a group of terms from a table representing the perceived ergonomic, hedonic and appeal quality aspects [22].

Due to the need of pre-prepared scenarios for some tasks, at the end of each evaluation, a database snapshot was stored for a possible future analysis and then the database and application were restored to the initial state.

The evaluation had 10 participants, a number reasonable enough to find most usability problems and perceive tendencies in users acceptance and preferences, with ages ranging from 21 to 44 years (24 mean value), with computer experience and a frequent use of internet. Most confirmed watching videos with frequency and occasionally use (less than one time per day) of movie information websites. Only one person had contact with MovieClouds in the past.

5.2 Tasks and Results

Results from the USE based evaluation are presented in table 5.1, by mean and standard deviation values for each feature, performed in the context of the 8 tasks, described next, followed by a global evaluation of SoundsLike.

In this section, each task delivered to users during the evaluation sessions is going to be described and the obtained results analysed. Observations made are also summarized.

In the first task, after reading the introductory sheet, users were presented with the application for the first time, with a random (non tagged) audio excerpt, and asked to identify every element that would represent the current audio excerpt in the interface. Here, most users were able to quickly identify all the components from the interface, even though some users had difficulties pointing to the current audio segment in the similarity graph, in the first contact.

The second task involved the playback of audio segments from the timeline and graph. Interviewers asked the users to play the sound and video of some excerpts. Most users identified a relationship between the graph nodes and the timeline elements and appreciated the interaction between every component during the video playback. Interviewers noticed that the quick sound playback while hovering a similar excerpt (T2.1) took an important role in the perception of those elements as audio segments by the users, without the use of additional informational tips. On the other hand, it was not obvious for some users in first time that clicking on every excerpt would play the associated video, although the feature was very much appreciated when learned.

The third task focused on the timeline features, such as the overview of the film provided by the video timeline, scrolling and manipulation of the dynamic zoom. Users were told to play the video audio excerpts in the zoom timeline in a fashion that would require them to manipulate the timeline to attain the task's objectives. All users used the scroll

buttons without problems, and the majority quickly perceived the zoom manipulation in the video timeline (T3.2). But most users did not find utility about the audio spectrogram (T3.3), which they reported was due to a lack of knowledge in the area of sound analysis, but recognized its importance to users that would have that knowledge.

The fourth task was meant to introduce the user to the real objective of the similarity graph and navigation by changing (selecting) the current audio excerpt. Interviewers simply asked the user to listen to the three most similar sounds, select one of them and observe the transition to become the current excerpt. The user's perception about each audio element and their colouring were also evaluated. It was observed most users got the distance metaphor correctly, but they did not move the graph to verify ambiguous cases (when every node stands almost at the same distance), until they got instructed to do so (T4.2) and since the distances did not differ that much, in this case the usefulness was 3.5.

		Usefulness		Satisfaction		Ease of Use	
Task	Feature	M	Δ	M	Δ	M	Δ
1.1	**Find the current node** in the **timeline**.	4.4	0.7	4.3	1.1	4.5	0.8
1.2	Find the current node in the **similarity graph**.	2.9	1.4	3.4	1.4	3.7	1.4
2.1	**Play** of the segment's sound.	3.7	1.1	4.2	0.8	4.4	0.7
2.2	Play of the segment's video.	4.7	0.5	4.6	0.7	4.4	0.8
3.1	**Movies Timeline**.	4.4	0.8	4.0	1.3	3.6	1.0
3.2	Zoom timeline.	4.3	1.1	4.2	1.1	4.3	0.9
3.3	Spectrogram timeline.	2.7	1.6	3.5	1.2	4.3	0.8
3.4	Timeline relationships.	4.6	0.5	4.4	0.7	4.0	0.9
3.5	Timeline - Overview.	4.6	0.5	3.9	0.7	3.8	0.8
4.1	**Similarity Graph**.	4.6	0.7	4.0	0.8	4.2	0.9
4.2	Graph dynamism.	3.5	1.2	3.6	0.8	2.5	1.0
4.3	Sound segments colours.	4.6	0.5	3.6	0.8	2.5	1.0
5.1	Choosing suggested **tags**.	4.6	0.5	4.4	0.5	4.3	0.9
5.2	Add new tags.	4.8	0.4	4.5	0.5	4.2	0.8
5.3	The possibility of tag rejection.	4.7	0.7	4.6	0.7	3.4	1.4
5.4	Adding tags fast.	5.0	0.0	4.4	1.0	2.4	1.4
6.1	**Play** of the segment's sound on **tagging context**.	4.8	0.4	4.8	0.4	4.6	0.5
6.2	Play of the segment's video on tagging context.	4.9	0.3	4.7	0.5	4.7	0.5
6.3	Using graph's **similar sounds** for tagging the current sound segment.	4.9	0.3	4.8	0.4	4.7	0.5
7	**In game context**, choosing the most similar is an efficient way of earning points?	4.5	1.3	4.4	1.1	4.5	1.0
8	**Points**' attribution for sound tagging	3.9	0.9	3.4	1.3	4.0	1.2
	SoundsLike Overall Evaluation	**4.4**	**0.5**	**4.2**	**0.6**	**3.9**	**0.6**
	Total (mean)	*4.3*	*0.6*	*4.2*	*0.4*	*4.0*	*0.6*

Table 5.1: USE Evaluation of SoundsLike (scale: 1-5). (M = Mean, Δ = Std. Deviation)

In the fifth task, labelling (tagging) is introduced to the users, where they could add some tags to audio excerpts and submit the changes to the database. Three cases were prepared: one audio excerpt without any kind of tag associated, and two with suggested tags: one case where tags were presented related with the current audio excerpt, the other one with unrelated tags. Interviewers noticed the users were able to introduce and accept suggestions (T5.1) without significant problems, but some failed to perceive the tag rejection feature (T5.3) without a proper explanation from the evaluator. Despite the usefulness of the fast tagging feature (comma separated text), without a proper tooltip, users were unable to find and use it without help, but it is a typical feature for more experienced users as a shortcut, very appreciated as soon they became aware of it.

With every user interface section covered, tasks 6, 7 and 8 were meant to immerse the user inside the labelling experience in the application.

The sixth task was meant to perceive how users would navigate back and forward in the application context and evaluate the similarity graph utility in this situation. Users were asked to listen to the current audio excerpt, then select a similar excerpt as the new current and to classify it immediately. Thereafter, users had the challenge to return to the previous one and repeat these actions again a couple of times. Interviewers observed users did not use the browser "back button" (which usually has "backspace" as shortcut) that is frequently used while browsing web pages. After inquiring the user about it, interviewers noticed this was caused due to the users immersion in the application, to the point they make no difference between a desktop and a web application, making necessary to display this feature inside the page (as a visible back button). Users found the similarity graph very useful and easy to use in a navigation context. Here, it was noticed that some users would sometimes double click to listen to audio excerpts, resulting in selecting the same audio as the current, one against the user will.

The eighth task was designed to give participants the most similar experience to a full SoundsLike game experience. Here users were asked to label ten audio excerpts without any restrictions, starting at a random excerpt chosen by the application. In this context, we observed that most users found the similarity graph the most useful relative to other components, due to the highly audio's similarity and the propagation of previous used labels as suggestions. By other words, users were attracted towards the classification of similar audio excerpts due to the higher change of getting good label suggestions and to their immediate availability in the interface (at a distance of just a glimpse of eye and a click). Here, interviewers observed that many users displayed engagement and immersion within the application, in less than four classified excerpts, and a increased number of compliments, which denotes a good acceptance, and a great quantity of suggestions to be discussed and applied in future development iterations.

Interviewers also inquired users about the scoring mechanism (Task 8 and Task 7), and they found it interesting and a great way to stimulate users to participate, although

they reported not having a full game experience in this short time of the evaluation. Users reported that gaining points provided a good feedback and motivation to keep up scoring, but most felt that this reward concept is not much effective due to the meaningless purpose of points in this prototype, because Gamification requires a constant adaptation to people's skill by adapting the rewards and game difficulty to keep players motivated (see section 2.4.4).

When inquired about the possibility of exchanging points for social and utility features (ranks, avatars, highlights, multiple user challenge, mini games, improved user interface tools, etc.) or for external services (movie tickets discounts, music streaming access), their recommended it as a strong motivation to use SoundsLike again. It is also interesting to note that more than half of the users displayed a great interest in rankings just for showing off and achievements, or competition. Fifty percent would also participate voluntarily, just for the sense of contributing.

5.3 Overall Evaluation

In the end, users were asked to rate SoundsLike globally. The values obtained were fairly high, with values of 4.4 for usefulness, 4.2 for satisfaction and 3.9 for ease of use, on average. This feedback offers a good stimulus for continuing the development and improvement of the application and project. The experience also allowed us to control and witness the rapid and fairly easy learning curve, and to notice that some of the least understood features in the first contact turned out to be the most appreciated.

Users provided us with a great amount of suggestions for improvements and for some possible new features. Engagement of most users was noticed when they were using the application freely to label excerpts without any restriction, during the execution of task 8. They pointed out the interface fluidity as one of the factors contributing to the engagement felt. The most appreciated features pointed out by the users, by descending order, were: the similarity graph, the timelines and the scoring system. The least appreciated was the audio spectrogram because some users commented on their lack of expertise in the field of audio analysis.

At the end of the interview, users were prompted to classify the application with the most relevant perceived ergonomic, hedonic and appeal quality aspects from [22] (8 positive and 8 negative terms for each category in a total of 48 terms), as many as they would feel appropriate.

Table 2 displays the most chosen terms, with the terms "Controllable" and "Original" on the top with 6 votes each, "Comprehensible", "Simple", "Clear" and "Pleasant" leading after with 5 votes each, followed by "Interesting", "Innovative", "Inviting" and "Motivating" with 4 votes, and "Supporting", "Complex", "Confusing" and "Aesthetic" with 3 votes.

#	Terms		#	Terms	
6	Controllable	H	4	Innovative	H
6	Original	H	4	Inviting	A
5	Comprehensible	E	4	Motivating	A
5	Simple	E	3	Supporting	E
5	Clear	E	3	*Complex*	E
5	Pleasant	A	3	*Confusing*	E
4	Interesting	H	3	Aesthetic	A

Table 5.2: Quality terms to describe SoundsLike. H:Hedonic; E:Ergonomic; A:Appeal [22]

Almost all the terms were positive, the most for Ergonomic qualities, which are related with traditional usability practices, efficiency, effectiveness and ease of use, and Hedonic quality, which comprises global and important dimensions with no direct relation tot he task the user wants to accomplish with the system. The two most frequent negative terms were “Complex” and “Confusing”, although the first is correlated with interesting, powerful or challenging applications (a frequent trade-off, also recognized in SoundsLike), and both terms are also opposite to the terms “Simple” and “Clear”, that were selected more often.

5.4 Perspectives

The evaluation provided us with a great quantity of feedback and the results obtained were quite good and encouraging. SoundsLike approaches and purposes got a amazing acceptance and users went through a pleasant experience where they benefit from more rich and precise information access while collaborate in the labelling task. Users found that the interface was original, controllable clear and pleasant. They particularly liked the similarity graph and the timeline representation. We were able to identify some usability aspects to improve. Despite the fact that this evaluation is only scratching the surface in terms of engagement and entertainment, we observed a good quantity of engagement in participants, it provided a great motivation for future work, and the use of richer interfaces may work in this context.

For future work, a detailed evaluation will be required to evaluate the effectiveness of current approaches providing engagement to users and other alternative solutions. This evaluation will need at least 20 people to obtain significant insights into the usefulness and usability of such approaches.

Chapter 6

Conclusions and Future Work

This chapter presents the final considerations about the developed work along the duration of this dissertation and summarizes some perspectives of work to be done in for a possible continuation.

6.1 Conclusion

The initial motivation for the developed work was based on extending MovieClouds with new features that would improve the browsing and exploration of movie spaces. However, the motivation changed when we realized that such system would require some mechanisms to extract information from large amounts of data of multimedia, and the inherent complexity difficulty in making it feasible. A new motivation was identified which shared some of the properties with the previous MovieClouds motivation.

A visualization to display differences between the content of audio excerpts has been designed and developed using a force-oriented graph and screen distance paradigm, and later reused as a utility to browse and classify audio excerpts in movies.

An approach to solve the cold start problem was presented which uses SoundsLike, a game with a purpose, to incite humans to participate collectively in the classification of movies soundtracks using simple textual labels as metadata and consensus to decide which suggested metadata will be more representative as classification/description for the associated sound excerpts. The same consensus will be used to calculate a proper reward for all users and work as incentive for a continuous classification. SoundsLike innovates with respect to previous games with a purpose with similar objectives by providing a rich context both in terms of temporal aspects through three timelines with different time scales, and in terms of similarities between items by displaying a dynamic force-directed graph where the neighbourhood of the current item is represented. SoundsLike interface lets the user play audio excerpts in severeal ways: by hovering with the pointer on the representation of the excerpt in the graph or by clicking it to play the correspondent video in the context of the movie.

Unlike most of the state of art, the application does not restrict the user to a closed set of labels but, by making suggestions, tends to overcome the never-ending vocabulary found in other labelling games. We could reduce the vocabulary even more by using natural language analysis and lexical databases (e.g. WordNet) to find relations between multiple labels and merging the most similar.

The developed work around SoundsLike was divided in three separated modules: the front-end, the web-service and the data storage. This architecture allows any application to be integrated within the system and access all the retrieved data through the web-service which provides a uniform and scalable interface with little implementation complexity (compared to existing alternatives). With this implementation, we are able to share all the results obtained from SoundsLike and other front-end applications, which are updated in real time, with almost no restrictions besides the indirect cost associated to the computational power and network traffic spent on hosting a server (or multiple servers, if replication is needed) over the internet.

The concept behind SoundsLike, as a crowd-sourcing human computation approach, was well received and accepted by users and scientific community, being published in three international conferences (see section 1.4). The evaluation results were quite good and encouraging, showing that the interface provides a joyful experience where users benefit from more rich and precise information access while they collaborate in the labelling task. Users found that the interface was original, controllable clear and pleasant, providing a encouraging results, useful feedback, and motivation for future work.

In addition, some work has been developed to extend MovieClouds and create to explore new representation of video and movies, however evaluations were not carried out on time for inclusion in this document. The proposed representations uses concentric wheel shapes to obtain a clock metaphor that mimics the way people may perceive passing of time to 1) overviews to represent the whole content of movies and their different perspectives with a stronger focus on visual properties, allowing to see how it changes along time; and 2) visualizations to compare movies in chosen perspectives of tracks of their contents (e.g scene images);

6.2 Future Work

The current state of work includes a diverse range of issues, features and tasks to be explored as future work. Some of these tasks can be executed in a near future, but others may require exploration and evaluations (e.g. the addition of game elements to keep users engaged).

The next steps to be taken, without any specific order, are:

1) The refinement of the interface based on feedback received and perceived issues, in the results obtained from the evaluation detailed on chapter 5, to address all the identi-

fied usability problems and improve the user interaction and perception while classifying audio towards an full a entertaining browsing experience.

2) The refinement of scoring and reward mechanisms and a better engagement-oriented evaluation. The current scoring mechanism is only based on points as the only reward, but users require some meaning and purpose. An improved evaluation is required to observe if points are rewarded fairly to the effort spent by users and the quality of the contributions, and which design options and features affects engagement. The idea is to provide ways to keep the user interested and motivated to contribute towards audio classification, even after leaving the application. There are many social and gaming elements to be explored, including: social network integration, challenges (single and multi-player challenges), charts, ranks, avatars, medals, achievements, extra or improved site features, and so on. Most of the features could be traded by the current currency we have available inside the game: points and time. I would not recommend the use of external currencies (real world money) without analysing the impact on players by adding a possible unfair advantage ready to be exploited by some players, which would have similar result as cheating has in most games.

3) Extend the navigation across different movies: Soundslike currently only supports the classification within a single movie, but it aims to classify any audio from any movie based on cross movie similarities. By allowing people to move from one movie to another, even without noticing it. We can immerse people in the entire movie spaces, maybe providing an experience more appealing and engaging compared to the current one restricted to random or continuous excerpts from the same movie. Here we are required to analyse if is important to distinguish excerpts from the same or others movies or to simply provide ways for users to restrict their navigation to a certain movie, genre, year, producer, etc.

4) Database improvements: The current database structure is dedicated to movie soundtrack labelling and only supports similarity relations between audio excerpts. An improved database is suggested in section 4.3.1 to store information for any multimedia document - music, movies, tv shows, etc. -, multiple types of relationships between media pieces and documents, and supports multimedia collections (e.g. tv series seasons, movies, music albuns, etc.) and authors (or artists).

5) Automatic Information Retrieval: the data retrieved through SoundsLike could be used to feed automatic information retrieval mechanisms based on statistical data models of information already classified. Data retrieved from automatic mechanisms could also be used to feed the same system and create a feedback cycle.

6) Improve content-driven recommendation systems, contextual advertisement and personalized retargeting: data retrieved could be mined to extract relationships useful for the cinematographic and discography industries. These relations could be used to improve existing recommendation systems by adding content based recommendations which may complete existing systems based on genres, global tags and users' activity. The same

relationships could be crossed with user information (activity, tastes, events) to produce custom advertisement for a specific user. A practical implementation for cinematographic industries would be using this information extracted from movies to produce custom trailers where the scenes could be chosen based on user preferences (i.e. a trailer focused on romance would pick more attention of users with a huge sentimental preferences than a trailer focused in war scenes).

7) Provide a "free" access to the database of audio classification to enhance content-based recommendation systems and sound analysis mechanisms.

References

[1] ALONSO, G., CASATI, F., KUNO, H., AND MACHIRAJU, V. Web Services. In *Web Services Concepts, Architectures and Applications*, no. Chapter 1. Springer Verlag, 2004, ch. 5.

[2] ARNOLD M. LUND. Measuring Usability with the USE Questionnaire. In *Usability and User Experience, 8(2).* Usability SIG, 2001.

[3] BAJEC, J., SMITS, D., SMID, H., HU, N., MEIJER, C., HONIG, M., SARDAR, A., AND VAN WIJK, N. Twitter Lyrics, 2010.

[4] BARRINGTON, L., O'MALLEY, D., TURNBULL, D., AND LANCKRIET, G. User-centered design of a social game to tag music. In *Proceedings of the ACM SIGKDD Workshop on Human Computation - HCOMP '09* (New York, New York, USA, June 2009), ACM Press, p. 7.

[5] BRAVE, S., AND NASS, C. Emotion in human-computer interaction. In *The human-computer interaction handbook*, J. A. Jacko and Andrew Sears, Eds. L. Erlbaum Associates Inc. Hillsdale, NJ, USA ©2003, Jan. 2002, pp. 81–96.

[6] BROWN, M. Comfort Zone: Model or metaphor? *Australian Journal of Outdoor Education 12*, 1 (2008), 3–12.

[7] CARD, S. K., MACKINLAY, J. D., AND SHNEIDERMAN, B., Eds. *Readings in information visualization: using vision to think.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[8] CHANG, C.-H., MEMBER, S., AND HSU, C.-C. Enabling concept-based relevance feedback for information retrieval on the WWW. *Knowledge and Data Engineering, IEEE Transactions 11*, 4 (1999), 595–609.

[9] CSIKSZENTMIHALYI, M. *Flow: The Psychology of Optimal Experience*, vol. 16. 1990.

[10] DANIEL, G., AND CHEN, M. Video visualization. In *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control* (Oct. 2003), IEEE, pp. 409–416.

[11] DETERDING, S., DIXON, D., KHALED, R., AND NACKE, L. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '11* (New York, New York, USA, Sept. 2011), ACM Press, p. 9.

[12] DETERDING, S., DIXON, D., SICART, M., NACKE, L., CULTURES, D., AND O'HARA, K. Gamification: Using Game Design Elements in Non-Gaming Contexts. In *CHI EA '11 Workshop* (New York, New York, USA, May 2011), ACM, pp. 2425–2428.

[13] EKMAN, P. Biological and cultural contributions to body and facial movement.

[14] EKMAN, P. Are there basic emotions? *Psychol Rev 99*, 3 (1992), 550–553.

[15] EKMAN, P. Chapter 3 Basic Emotions. In *Handbook of Cognition and Emotion*, no. 1992. 1999.

[16] FEW, S. Data Visualization - Past, Present, and Future. *IBM Cognos Innovation Center* (2007).

[17] FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.

[18] FLANAGAN, D., AND FERGUSON, P. *JavaScript: The Definitive Guide*, 5 ed. O'Reilly & Associates, 2006.

[19] GAULIN, S. J. C., AND MCBURNEY, D. H. *Evolutionary psychology*. Pearson-/Prentice Hall, 2004.

[20] GIL, N., SILVA, N., DIAS, E., MARTINS, P., LANGLOIS, T., AND CHAMBEL, T. Going Through the Clouds: Search Overviews and Browsing of Movies. In *Proceeding of the 16th International Academic MindTrek Conference* (Tampere, Finland, 2012), ACM, pp. 158–165.

[21] GOMES, J. M. A., CHAMBEL, T., AND LANGLOIS, T. SoundsLike: Movies Soundtrack Browsing and Labeling Based on Relevance Feedback and Gamification. In *Proceedings of the 11th european conference on Interactive TV and video* (Como, Italy, 2013), ACM, pp. 59—-62.

[22] HASSENZAHL, M., PLATZ, A., BURMESTER, M., AND LEHNER, K. Hedonic and ergonomic quality aspects determine a software's appeal. *Proceedings of the SIGCHI conference on Human factors in computing systems CHI 00 2*, 1 (2000), 201–208.

[23] HAUPTMANN, A. G. Lessons for the future from a decade of informedia video analysis research. In *Conference of Image and Video Retrieval* (Singapure, July 2005), W.-K. Leow, M. S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, and E. M. Bakker, Eds., vol. 3568 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 1–10.

[24] HAYKIN, S. *Advances in spectrum analysis and array processing (vol. III)*. Prentice-Hall, Inc., May 1995.

[25] HERGENHAHN, B. *Introduction to the History of Psychology*. Wadsworth Publishing Company, 2008.

[26] HIEMSTRA, D., AND ROBERTSON, S. Relevance Feedback for Best Match Term Weighting Algorithms in Information Retrieval. In *Dublin City University* (2001), pp. 37–42.

[27] HUOTARI, K., AND HAMARI, J. Defining Gamification - A Service Marketing Perspective. *MindTrek 2012* (2012).

[28] INC., N. Netflix Investor Relations Overview. `http://ir.netflix.com/`.

[29] JAFARINAIMI, N. Exploring the Character of Participation in Social Media: The Case of Google Image Labeler. In *iConference '12* (Toronto, Canada, 2012), ACM.

[30] JING, F., ZHANG, M. L. H.-J., AND ZHANG, B. Learning Region Weighting From Relevance Feedback In Image Retrieval. *IEEE ICASSP'02 Vol: 4* (2002), 4088–4091.

[31] JORGE, A., AND CHAMBEL, T. Exploring Movies through Interactive Visualizations. In *27th International British Computer Society Human Computer Interaction Conference* (2013), p. 6.

[32] JORGE, A., GIL, N., AND CHAMBEL, T. Time for a New Look at the Movies through Visualization. In *6th International Conference on Digital Arts* (Faro, Portugal, 2012), T. Chambel, A. G. Ariza, G. Perin, M. Tavares, J. Bidarra, and M. Figueiredo, Eds., Artech International, pp. 269–278.

[33] KESTERENM, A. V., AND PIETERS, S. HTML5 differences from HTML4. `http://www.w3.org/TR/2011/WD-html5-diff-20110405/`, 2013.

[34] KHALED, R. It's Not Just Whether You Win or Lose: Thoughts on Gamification and Culture. In *Gamification Workshop at ACM CHI'11* (2011), pp. 1–4.

[35] LAW, E. Defining (Human) Computation. *CHI'11 Workshop on Crowdsourcing and Human Computation* (2011).

[36] LEVY, M. How are you feeling today? - Last.fm Blog, 2012.

[37] LEVY, M., AND SANDLER, M. A semantic space for music derived from social tags. *ISMIR 2007* (2007).

[38] LIKERT, R. A technique for the measurement of attitudes. *Archives of Psychology 22* (1932), 5–55.

[39] LIU, M., AND WAN, C. Weight Updating For Relevance Feedback In Audio Retrieval. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference* (2003), vol. d, pp. 644–647.

[40] LUCKNER, J. L., AND NADLER, R. S. *Processing the experience : strategies to enhance and generalize learning*, 2 ed. Dubuque, Iowa: Kendall/Hunt, 1997.

[41] MANDEL, M., ELLIS, D., AND MICHAEL I MANDEL, D. P. W. E. A web-based game for collecting music metadata. *Journal of New Music Research 37*, 2 (2008), 15.

[42] MARTINHO, J. A., AND CHAMBEL, T. ColorsInMotion: Interactive Visualization and Exploration of Video Spaces. In *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era on - MindTrek '09* (New York, New York, USA, Sept. 2009), ACM Press, p. 190.

[43] MASLOW, A. H. A Theory of Human Motivation. *Psychological Review 50*, 4 (1943), 370–396.

[44] MAUCH, M., AND LEVY, M. Structural Change On Multiple Time Scales As A Correlate Of Musical Complexity. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)* (2011), pp. 489–494.

[45] MCLEOD, S. A. B.F. Skinner | Operant Conditioning - Simply Psychology. `http://www.simplypsychology.org/operant-conditioning.html`, 2007.

[46] MORTON, B. G., SPECK, J. A., SCHMIDT, E. M., AND KIM, Y. E. Improving music emotion labeling using human computation. In *Proceedings of the ACM SIGKDD Workshop on Human Computation - HCOMP '10* (New York, New York, USA, July 2010), ACM Press, p. 45.

[47] NEPUSZ, T. Reconstructing the structure of the world-wide music scene with Last.fm. `http://sixdegrees.hu/last.fm/`, 2008.

[48] NETCRAFT. December 2012 Web Server Survey. http://news.netcraft.com/archives/2012/12/04/december-2012-web-server-survey.html, 2013.

[49] OBRIEN, O. Emotionally Vague, 2006.

[50] OLIVEIRA, E., BONOVOY, M., RIBEIRO, N., AND CHAMBEL, T. Towards Emotional Interaction: Using Movies to Automatically Learn Users' Emotional States. In *INTERACT' 2011, 13th IFIP TC13 Conference on Human-Computer Interaction* (Lisboa, Portugal, 2011).

[51] OLIVEIRA, E., MARTINS, P., AND CHAMBEL, T. iFelt : Accessing Movies Through Our Emotions. In *9th International Conference on Interactive TV and Video: Ubiquitous TV* (2011), pp. 105–114.

[52] PLUTCHIK, R. The Nature of Emotions. *American Scientist, Volume 89* (2001).

[53] PRESS, T. A. By The Numbers: Breakdown of Netflix's subscribers as of June 30. http://news.yahoo.com/numbers-netflix-subscribers-205626746.html, July 2013.

[54] RADICAT, S., AND BUCKLEY, T. Email Market, 2012 - 2016. http://www.radicati.com/wp/wp-content/uploads/2012/10/Email-Market-2012-2016-Executive-Summary.pdf, October 2012.

[55] RUSSELL, J. A. A circumplex model of affect. *Journal of Personality and Social Psychology 39*, 6 (1980), 1161–1178.

[56] SCHACTER, D., L., D., GILBERT, T., AND WEGNER, D. M. Chapter 7.9 B. F. Skinner: The Role of Reinforcement and Punishment. In *Psychology. ; Second Edition*. Worth, Incorporated, 2011, ch. 7.9.

[57] SKINNER, B. F. *The Behavior of Organisms*. Copley Publishing Group, 1938.

[58] TAMAS, N. Reconstructing the structure of the world-wide music scene with last.fm, 2008.

[59] TUFTE, E. R. *Beautiful Evidence*. Graphics Press, 2006.

[60] TURNBULL, D., LIU, R., BARRINGTON, L., AND LANCKRIET, G. Using Games To Collect Semantic Annotations Of Music. *ISMIR 2007* (2007), 1–3.

[61] TURNBULL, D. R. Design and Development of a Semantic Music Discovery Engine A. University of California.

[62] VERISIGNINC. The Domain Name Industry Brief. http://www.verisigninc.com/assets/domain-name-brief-dec2012.pdf, December 2012.

[63] VON AHN, L., BLUM, M., HOPPER, N. J., AND LANGFORD, J. CAPTCHA: Using Hard AI Problems for Security. In *EUROCRYPT 2003* (2003), E. Biham, pp. 294–311.

[64] VON AHN, L., AND DABBISH, L. Labeling images with a computer game. In *CHI '04* (New York, New York, USA, 2004), vol. 6, ACM, pp. 319–326.

[65] VON AHN, L., KEDIA, M., AND BLUM, M. Verbosity: a game for collecting common-sense facts. In *Knowledge Creation Diffusion Utilization* (2006), R. E. Grinter, T. Rodden, P. M. Aoki, E. Cutrell, R. Jeffries, and G. M. Olson, Eds., vol. 1, ACM, ACM Press, pp. 75–78.

[66] VON AHN, L., LIU, R., AND BLUM, M. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06* (New York, New York, USA, Apr. 2006), ACM Press, p. 55.

[67] W3C. HTML & CSS. http://www.w3.org/standards/webdesign/htmlcss#whatcss.

[68] WAN, C., AND LIU, M. Content-based audio retrieval with relevance feedback. *Pattern Recognition Letters 27*, 2 (Jan. 2006), 85–92.

[69] WANG, R., NAPHADE, M., AND HUANG, T. Video retrieval and relevance feedback in the context of a post-integration model. *2001 IEEE Fourth Workshop on Multimedia Signal Processing (Cat. No.01TH8564)* (2001), 33–38.

[70] WU, M. Gamification 101: The Psychology of Motivation . http://lithosphere.lithium.com/t5/science-of-social-blog/Gamification-101-The-Psychology-of-Motivation/ba-p/21864, 2011.

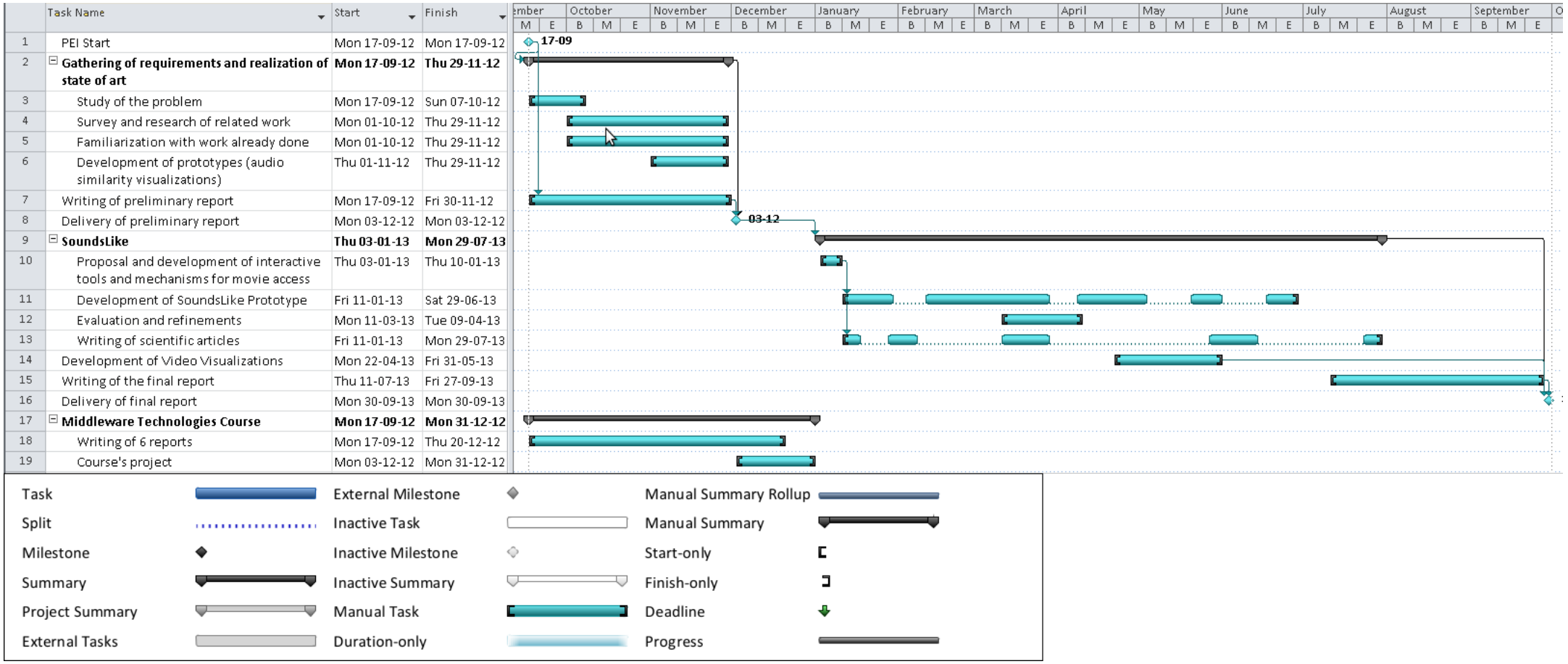
[71] WWW.DB ENGINES.COM. Database Engines Ranking. http://db-engines.com/en/ranking.

[72] WWW.INTERNETWORLDSTATS.COM. World Internet Users Statistics Usage and World Population Stats. http://www.internetworldstats.com/stats.htm, 2012.

[73] WWW.PHP.NET. History of PHP. http://www.php.net/manual/en/history.php.php.

Appendix A

Final Gantt Map



	Task Name	Start	Finish
1	PEI Start	Mon 17-09-12	Mon 17-09-12
2	**Gathering of requirements and realization of state of art**	**Mon 17-09-12**	**Thu 29-11-12**
3	Study of the problem	Mon 17-09-12	Sun 07-10-12
4	Survey and research of related work	Mon 01-10-12	Thu 29-11-12
5	Familiarization with work already done	Mon 01-10-12	Thu 29-11-12
6	Development of prototypes (audio similarity visualizations)	Thu 01-11-12	Thu 29-11-12
7	Writing of preliminary report	Mon 17-09-12	Fri 30-11-12
8	Delivery of preliminary report	Mon 03-12-12	Mon 03-12-12
9	**SoundsLike**	**Thu 03-01-13**	**Mon 29-07-13**
10	Proposal and development of interactive tools and mechanisms for movie access	Thu 03-01-13	Thu 10-01-13
11	Development of SoundsLike Prototype	Fri 11-01-13	Sat 29-06-13
12	Evaluation and refinements	Mon 11-03-13	Tue 09-04-13
13	Writing of scientific articles	Fri 11-01-13	Mon 29-07-13
14	Development of Video Visualizations	Mon 22-04-13	Fri 31-05-13
15	Writing of the final report	Thu 11-07-13	Fri 27-09-13
16	Delivery of final report	Mon 30-09-13	Mon 30-09-13
17	**Middleware Technologies Course**	**Mon 17-09-12**	**Mon 31-12-12**
18	Writing of 6 reports	Mon 17-09-12	Thu 20-12-12
19	Course's project	Mon 03-12-12	Mon 31-12-12

Appendix B

SoundsLike Installation Manual

B.1 Install Apache+Mysql+PHP (for Linux Debian based distros)

Note: superuser access is required!

1. Download XAMPP package installer for linux (from http://apachefriends.org);
   Note: as alternative you can install apache2, mysql and php packages from the repositories but you will need a lot more configurations!

2. Run the installer and follow the instructions;

   > chmod 755 xampp-linux-X.X.X-X-installer.run

   > ./xampp-linux-X.X.X-X-installer.run

3. Start XAMPP. If you already have an application running in port 80, jump to step 4.

   > sudo /opt/lampp/lampp start

4. Change apache configuration, including the listening port;
   Edit the file "/opt/lampp/etc/httpd.conf" with super user;
   Important config entries:
   Listen <port> - listening port.
   ServerRoot <location> - server location (do not change it!).
   ServerAdmin <email> - server admin email.
   ServerName <name> - domain or ip. Usually guessed automatically.
   DocumentRoot <location> - web files location, by default in "/opt/lampp/htdocs".

5. Improve security:

   > sudo /opt/lampp/lampp security

IMPORTANT!!!! —- Take note of all inserted passwords, including MySQL root password. Do not leave mysql root user with an empty password!

6. Test: http://localhost/

B.2 XAMPP Command Parameters

XAMPP command:

> sudo /opt/lampp/lampp [parameters]

> **example:** sudo /opt/lampp/lampp restart

XAMPP parameters:

Parameter:	Description:
start	Starts XAMPP.
stop	Stops XAMPP.
restart	Stops and starts XAMPP.
startapache	Starts only the Apache.
startssl	Starts the Apache SSL support. This command activates the SSL support permanently, e.g. if you restarts XAMPP in the future SSL will stay activated.
startmysql	Starts only the MySQL database.
startftp	Starts the ProFTPD server. Via FTP you can upload files for your web server (user "nobody", password "lampp"). This command activates the ProFTPD permanently, e.g. if you restarts XAMPP in the future FTP will stay activated.
stopapache	Stops the Apache.
stopssl	Stops the Apache SSL support. This command deactivates the SSL support permanently, e.g. if you restarts XAMPP in the future SSL will stay deactivated.
stopmysql	Stops the MySQL database.
stopftp	Stops the ProFTPD server. This command deactivates the ProFTPD permanently, e.g. if you restarts XAMPP in the future FTP will stay deactivated.
security	Starts a small security check program.

B.3 Important Files And Directories (XAMPP)

File/Directory :	Purpose:
File/Directory	Purpose
/opt/lampp/bin/	The XAMPP commands home. /opt/lampp/bin/ calls for example the MySQL monitor.
/opt/lampp/htdocs/	The Apache DocumentRoot directory.
/opt/lampp/etc/httpd.conf	The Apache configuration file.
/opt/lampp/etc/my.cnf	The MySQL configuration file.
/opt/lampp/etc/php.ini	The PHP configuration file.
/opt/lampp/etc/proftpd.conf	The ProFTPD configuration file. (since 0.9.5)
/opt/lampp/phpmyadmin/ config.inc.php	The phpMyAdmin configuration file.

B.4 First Steps for SoundsLike Instalation

Note: I am assuming that the **DocumentRoot** is **"/opt/lampp/htdocs" Note2:** If you dislike doing file operation in Linux terminal, run one of the following commands to obtain a graphical file manager with super user permissions:

> (Gnome[1]) gksudo nautilus
> (Mate[2]) gksudo caja
> (other) gksudo <yourFileManagerNameHere>

Obtain the virus webservice package, soundslike frontend package and database SQL file!

B.5 Installing VIRUS Database

1. Go to http://localhost/phpmyadmin/ ;
2. Use “root” as username and the respective password set previously on the security step (if you did not change it, the default is empty. For this case I advice you to change it now!);
3. Create a new database called “virus-data”;
4. Go to “Users” tab;
5. Add a new database user called “virus”, host “localhost”, a custom password and NO custom privileges;
6. Edit “virus” user privileges, in the “Database-specific privileges” add access to the database "virus-data" by using the available drop-box and give all privileges (or just

[1]Gnome Desktop Manager, e.g. Linux Ubuntu)
[2]Mate Desktop Manager, e.g. Linux Mint)

"data" previleges);
NOTE: If you want a improved security, only give "Data" manipulation privileges and user root for changing database structure and other administrative changes.

7. Select database "virus-data" in the left panel displaying a database list.
8. Go to the "Import" tab;
9. Select the database SQL file by clicking the "browse" button and click "Go";
Note: in case of error due size file, try compress to a zip and import, otherwise you will have to use mysql console or a software similar to Navicat.
10. Database configuration done. Take note of the user and password for this database.

B.6 Installing VIRUS Web-service

1. Obtain the virus webservice package;
2. Unpack the virus webservice package and move the contents of the folder "webservice" to "/opt/lampp/htdocs/virus-webservice";
3. Edit "/opt/lampp/htdocs/virus-webservice/index.php".
Find the configuration array and change the following entries:
'dbUser' => 'virus',
'dbPassword' => '<YourDatabasePasswordConfiguredBefore>',
'dbName' => 'virus-data',
NOTE: later I advice you to change "debug" to false. The webservice will log everything to "/opt/lampp/htdocs/virus-webservice/logs/"
4. Open http://localhost/virus-webservice/
5. If displays error relative the opening of log files run the following commands:
   > sudo chgrp -R nogroup /opt/lampp/htdocs/
   > sudo chmod -R 770 "/opt/lampp/htdocs/virus-webservice/"
   Note: Apache process operates under the user "nobody" and group "nogroup". You need give proper permissions for Apache being able to create files inside htdocs.
6. Test again. In case of error, check the permissions again (or use chmod 777 on "logs" directory, but introduces a security fault)
7. Test "http://localhost/virus-webservice/index.php/apiv1/video"
8. Webservice configuration done!

B.7 Installing SoundsLike Front-end

1. Obtain the Soundslike front-end package;
2. Unpack the Soundslike package and move the contents of the folder “public_html” to “/opt/lampp/htdocs/soundslike”;
3. Edit “/opt/lampp/htdocs/soundslike/index.html”.
   Find “var api = new MovieCloudsApi(“ and add the web-service url.
   For this manual instructions, the web-service configuration URL is “http://localhost/virus-webservice/index.php/”

Appendix C

VIRUS Webservice Documentation - API Version 1

C.1 Base URI

All requests and responses coming for and from the API are sent and HTTP messages. The request includes a URI to indicate what service and collection are desired, and HTTP headers to provide context for the request which may affect directly or indirectly the response.

The Web Service uses a Rest-full API and supports the following HTTP methods: GET, POST, PUT and DELETE.

These methods have their meanings described in the HTTP/1.1 specification (RFC 2616): "The GET method means retrieve whatever information is identified by the Request-URI. The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. The PUT method requests that the enclosed entity be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource, the enclosed entity SHOULD be considered as a modified version of the one residing on the origin server. The DELETE method requests that the origin server delete the resource identified by the Request-URI."

The web-service is accessible by using the following generic URI path:
<base-url>/apiv1/[<resource>/[<resource-id>[/<resource-assoc>]]][/[<parameter>]*]

Where:

- the **HTTP Method** field specifies the operation type. The API only supports GET, POST, PUT and DELETE, and access to these methods may change due to restric-

tions imposed by the service (resource) that should be documented (e.g. authentication may be required). If not specified or invalid, GET is used by default;

- the **HTTP Accept** field specifies the content types acceptable for the response. The API only supports “xml” and “json” accept types. If not specified or invalid, “xml” is used by default;
- the **HTTP Content-Type** field specifies the MIME type of the body of the request (used with POST and PUT requests). The API only supports “xml” and “json” accept types. If not specified or invalid, “xml” is used by default;
- **<base-url>** is the base URI for accessing the web-service API instance.
  Example: http://www.myamazingdomain.com/virus-webservice/webservice
- **<resource>** Also known as service, from the developers point of view. Resource is the wanted resource collection that you desire to access.
  E.g. “GET apiv1/video” returns a list of available movies, while GET “apiv1/users” returns a list of returns a list of users;
- **<resource-id>** an specific id for operations with a specific <resource> collection entity or entry.
  E.G. “GET /api/video/1234” returns the video entry with the id “1234”, while “DELETE /api/user/10” removes the user entity with id “10”;
- **<resource-assoc>** associated <resource>. This collection is filtered by the entity <entryId>.
  E.g. “GET /user/1/soundtags" should get a list of tags filtered by the user with id “1”.
- **<parameter>** additional request parameters in the format “key:value” or “key=value”. Parameters may be global and available in the entire API, or local available in some services (resources) or associated resources.
  Global parameters:
  - limit - limits operations to a number of operations. For GET results, represents the number of entities returned by page. Results are limited to a default value of 100 entries;
  - page - for GET results, represents the page number for returned results. By default the value is set as 1 (the first page of results).

C.2 Resource Collections

The following collection associations are available:

- /video/<video-id>/
  - soundsegment - Collection of audio segments associated to a selected <video-id>.
- /video/<video-id>/soundsegment/<segment-id>
  - similar - Collection of similar audio segments related to a selected audio segment, ONLY inside the selected video. For selecting similar segments in other videos, use the /soundsegment/ service.
  - soundtag - Collection of tags associated to the selected audio segment with <segment-id> id.
- /soundsegment/<segment-id>/
  - similar - Collection of similar audio segments related to a selected audio segment. For selecting similar segments inside a single video, use the /video/ service.
  - soundtag - Collection of tags associated to the selected audio segment with <segment-id> id.
- /soundsegment/<segment-id>/soundtag/ or /video/<video-id>/soundsegment/<segment-id>/ user/<user-id> - Filters the soundtag collection by the specified <user-id>.
- /user/<user-id>/
  - soundtag - Collection of all inserted tags by the user with id <user-id>, ordered by inserted time.

C.3 Errors

Any error returned to the client are identified by the HTTP response error code. The response content body will also contains the error information in xml or json format. Table C.1 displays a short example of a returned error returned by the web-service when the user do not specifies an existing API name and version in the request URI.

All responses returned by the webservice with HTTP codes that starts by 4xx and 5xx, must be treated as errors: the first (4xx) are caused by errors introduced by the client, the second (5xx) by errors in the server.

List of errors and default descriptions:

400 Invalid Parameter: Your request is missing a required parameter;

```
<virus status="404" code="404 Not Found">
<error>
<code>404</code>
<title>API Not Found</title>
<description>Please specify a valid API and Service!</description>
</error>
</virus>
```

Table C.1: A content body example for a error response from the web-service, in XML.

401 , **402** and **403** Authentication Failed: You do not have permissions to access the service;

404 Invalid Resource: This resource does not;

405 Invalid Method: No method with that name in this service;

409 Conflict: Your request is conflicting with data present in the server;

418 Iḿ a teapot: Iḿ a little teapot, Short and stout, Here is my handle (one hand on hip), Here is my spout (other arm out with elbow and wrist bent), When I get all steamed up, Hear me shout, Tip me over and pour me out (lean over toward spout)!;

429 Too Many Requests: You have reached your request quota per client. Try again later;

500 Operation Failed: Something went wrong with the server;

501 Not Implemented: The service/method you were trying to reach is not implemented here;

503 Invalid Resource: This service is temporarily offline. Try again later;

Appendix D

User Evaluation Script

Plano de Avaliação V3
SoundsLike V1

#

Idade: ____
Gênero:
[ ] Masculino; [ ] Feminino;
Frequência de uso de Internet:
[ ] Nunca; [ ] Raramente; [ ] Ocasionalmente; [ ] 1 vez por dia; [ ] Diversas vezes por dia;
Experiência na área de informática:
[ ] Baixa; [ ] Media; [ ] Avançada;
Costuma visualizar filmes e vídeos na internet:
[ ] Nunca; [ ] Raramente; [ ] Ocasionalmente; [ ] 1 vez por dia; [ ] Diversas vezes por dia;
Costuma visitar sites com informações de filmes ou vídeos (e.g. imdb.com):
[ ] Nunca; [ ] Raramente; [ ] Ocasionalmente; [ ] 1 vez por dia; [ ] Diversas vezes por dia;
Já alguma vez experimentou o MovieClouds no passado?:
[ ] Sim; [ ] Não;

Texto introdutório:

"Encontra-se a experimentar o SoundsLike, uma aplicação interactiva de classificação de excertos de áudio recorrendo a etiquetas textuais fornecidas por diversos utilizadores.
Esta aplicação é dada no contexto do MovieClouds, uma aplicação web interactiva onde os utilizadores podem aceder, explorar e visualizar filmes com base na informação obtida do seu conteúdo, especialmente áudio e legendas, estando focada especialmente nas dimensões emocionais expressas pelo filme ou sentida pela assistência.

A categorização automática de conteúdos de áudio é uma tarefa significantemente árdua, e abordagens existentes baseiam-se essencialmente em modelos estatísticos que geralmente em torno de etiquetas textuais. A existência de base de dados com este tipo de informação é escassa e a construção implica um trabalho imenso que requer horas a escutar áudio e classificação manual.

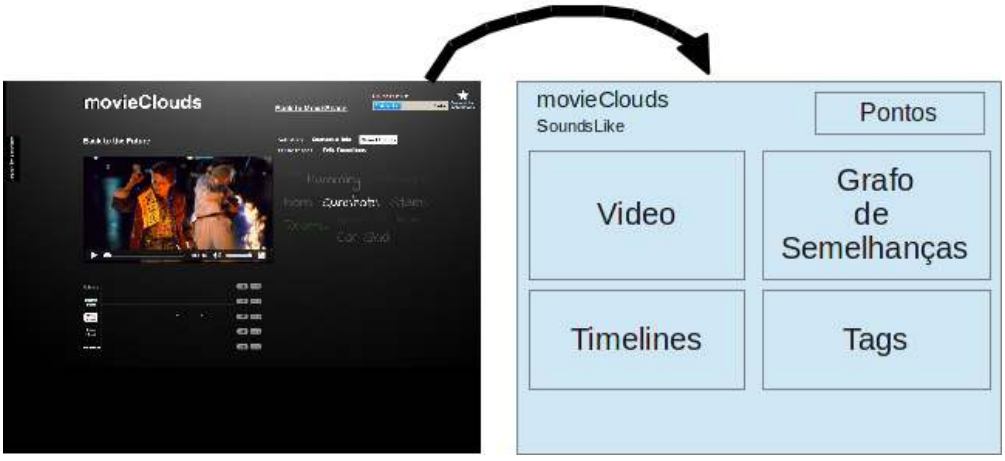
O SoundsLike trata-se de uma jogo aonde os jogadores participam voluntariamente na categorização do áudio de filmes, num esquema de crowdsourcing, com o objectivo de se recolher informação sobre o áudio de filmes e premiar os utilizadores pela concordância das etiquetas dadas entre si, ao mesmo tempo que se obtêm dados para avaliar algoritmos de classificação de áudio. As contribuições são simplemente realizadas através de etiquetas textuais com o objectivo de catalogar o conteúdo dos excertos de áudio.

Você encontrava-se a utilizar a aplicação MovieClouds para visualizar o filme "Return to the Future", e decidiu experimentar o SoundsLike. Está a utilizar neste momento uma conta de utilizador pré-definida para esta avaliação, que foi já utilizada anteriormente. Irá se deparar com alguns excertos de áudio já "classificados por si".

Cada excerto de áudio possui 4 segundos de comprimento, tendo sido retirado do mesmo filme.
Neste momento a aplicação escolheu aleatoriamente um excerto de áudio do filme como ponto de partida para as tarefas que o avaliador lhe irá propor a partir de agora."

Por favor diga tudo o que estiver a pensar ao testar a aplicação!

A figura em baixo resume os componentes principais do interface do SoundsLike:



Tarefa 1: Identificar o áudio actual no grafo e timeline

- **Que pré condições são necessárias (contexto)?**
  - Foi escolhido um excerto de áudio aleatoriamente, que se encontra por classificar.
- **Que funcionalidades estão em causa?**
  - Timeline - Estado - áudio actual
  - Grafo - Estado - áudio actual.
- **O que avaliar?**
  - Facilidade de identificação dos componentes que integram a timeline;
  - Contexto e utilidade dos elementos;
- **<u>O que pedir?</u>**
  1. Identifique o excerto de áudio actual? Em que mais outros locais pode encontrar o áudio actual representado?
- **O que observar?**
  - Conseguiu identificar o excerto actual rapidamente:
    - Timeline (S/N): _____
    - Grafo (S/N): ____
    - Espectograma (S/N): ____
  - Outras observações:

- **O que perguntar?**
  - Foi obvio ou fácil identificar o áudio actual? (1-5) _____
  - No contexto de classificação de áudio, classifique...
    - a Timeline:
      - A utilidade (1-5) : ____
      - A satisfação (1-5): ____
      - Facilidade (1-5): ____
    - o Grafo:
      - A utilidade (1-5) : ____
      - A satisfação (1-5): ____
      - Facilidade (1-5): ____
  - Opinião?

Tarefa 2: Reprodução do excerto sonoro e video.

- **pré condições são necessárias (contexto)?**
  - Foi escolhido um excerto de áudio aleatoriamente, que se encontra por classificar.
- **Que funcionalidades estão em causa?**
  - Timeline - Onde se encontra actualmente ;
  - Correspondência entre elementos do grafo, timeline e video;
  - 2 níveis de interacção (timeline e grafo): hover e click.
- **O que avaliar?**
  - Utilização dos elementos;
  - Contexto e utilidade dos elementos;
- **<u>O que pedir?</u>**
  1. Reproduzir o áudio ou video associado ao excerto actual.
  2. Passar com o rato em cima para ouvir o áudio actual. Agora clicar no mesmo áudio.
  3. Experimentar o mesmo em outros áudios, usando a timeline e o grafo.
- **O que observar?**
  - Conseguiu (sozinho) identificar 2 modos de reprodução de áudio? (1-3) ____
  - Referiu a interacção entre timeline e grafo? (S/N) ___
  - Outras observações:

- **O que perguntar?**
  - Foi obvio perceber como reproduzir o áudio? (1-5) ____
  - Foi obvio perceber como reproduzir o video? (1-5) ____
  - Áudio (Objectivo classificar o áudio):
    - Utilidade: (1-5) ___
    - Facilidade: (1-5) ___
    - Satisfação: (1-5) ___
  - Video (Objectivo classificar o áudio):
    - Utilidade: (1-5) ___
    - Facilidade: (1-5) ___
    - Satisfação: (1-5) ___
  - Opinião/sugestões?

Tarefa 3: Navegação da Timeline

- **Que pré condições são necessárias (contexto)?**
  - nenhuma
- **Que funcionalidades estão em causa?**
  - Estado (significado das cores, áudio actual)
  - Continuidade/Sequência de excertos de áudio
    - Possibilidade de visualizar áudios contíguos temporalmente
    - Possibilidade de navegar para outras partes do filme.
  - Zoom
    - Zoom dinâmico: aumentar e diminuir área de zoom
    - Movimentar da área de zoom dinâmica
  - 3 níveis de timeline
    - Utilidade dos diferentes níveis
    - Relação entre os diferentes níveis
    - Identificação dos diferentes níveis de interacção com os nós:
- **O que avaliar?**
  - Utilidade da timeline para a navegação entre excertos sonoros de um video.
- **<u>O que pedir</u>**
  1. Pedir para reproduzir 3 áudios contíguos ao actual (1 anterior, 2 seguintes);
  2. Pedir para reproduzir o 8º áudio depois do actual
  3. Pedir para reproduzir um áudio no inicio ou final no video.
  4. Voltar a visualizar o áudio actual na timeline e identificar a mesma região nas diferentes timelines.
- **O que observar?**
  - Qual o método utilizado no passo 2, 3, 4 para navegação (Zoom, Scroll) (Zoom, Scroll) _____ _____ _____
  - O utilizador indentificou as regiões actuais em cada timeline? (1-5) _____
  - Outras observações:

- **O que perguntar?**
  -
  - Timeline do filme todo:
    - Utilidade: (1-5) ____ Facilidade: (1-5) ____ Satisfação: (1-5) ____
  - Timeline com zoom dinâmico:
    - Utilidade: (1-5) ____ Facilidade: (1-5) ____ Satisfação: (1-5) ____
  - Timeline de espectro de audio - Espectograma:
    - Utilidade: (1-5) ____ Facilidade: (1-5) ____ Satisfação: (1-5) ____
  - Relação entre elementos (e identificar regiões actuais em cada timeline):
    - Utilidade: (1-5) ____ Facilidade: (1-5) ____ Satisfação: (1-5) ____
  - Avaliação global da timeline (como um todo):
    - Utilidade: (1-5) ____ Facilidade: (1-5) ____ Satisfação: (1-5) ____
  - O que acha do modo de manipulação e controlo do zoom da timeline? (descritivo)
  - Opinião / Sugestões?

Tarefa 4: Identificação do áudio mais semelhante no grafo.

- **Que pré condições são necessárias (contexto)?**
    - O áudio actual terá que ter relações de semelhança variadas com outros excertos de áudio, de preferência com elementos já etiquetados.
- **Que funcionalidades estão em causa?**
    - Todas as funcionalidades do Grafo de semelhanças excepto double click em nós.
- **O que avaliar?**
    - Avaliar a utilidade e percepção do grafo para encontrar e analisar rapidamente excertos de áudio semelhantes.
- **<u>O que pedir?</u>**
    1. Identificar e reproduzir os 3 áudios mais semelhantes.
    2. Desses 3, focar o que considera mais semelhante (double click).
    3. Observar as cores dos elementos (Etiquetado, Skipped, Por anotar)
- **O que observar?**
    - Conseguiu imediatamente apontar o áudio mais semelhante? (S/N) ___
    - Manipulou o grafo para conseguir reduzir as diferenças? (S/N) ____
    - Outras observações:

- **O que perguntar?**
    - É imediatamente perceptível que as distancias entre os nós representam a similaridade entre os mesmos? (S/N) ___
    - Representação no grafo de áudios semelhantes:
        - Utilidade: (1-5) ____ Facilidade: (1-5) ____ Satisfação: (1-5) ____
    - Dinamismo do grafo (arrastar dos nós):
        - Utilidade: (1-5) ____ Facilidade: (1-5) ____ Satisfação: (1-5) ____
    - Cores dos elementos:
        - Utilidade: (1-5) ____ Facilidade: (1-5) ____ Satisfação: (1-5) ____
    - Opiniões/Sugestões:

Tarefa 5: Categorizar o áudio actual

- **Que pré condições são necessárias (contexto)?**
  - Necessários 3 casos pre selecionados: 1. áudio sem tags atribuídas, 2. áudio com tags já atribuídas ou com vizinhos com tags atribuídas (tags boas) e 3. tags más.
- **Que funcionalidades estão em causa?**
  - Inserção de novas etiquetas
    - Inserção de múltiplas etiquetas
  - Sugestão de etiquetas
  - Diferentes estados das etiquetas (ignorar, aceitar, rejeitar)
  - Diferença entre etiquetas e estados (cores & ícones dos estados, etiquetas sugeridas vs inseridas)
  - Atribuição de pontos
    - Como são atribuídos e como incitam a participação com o utilizador
- **O que avaliar?**
  - Utilidade e usabilidade do interface de etiquetagem e modo de como as sugestões de etiquetas são apresentadas.
- **<u>O que pedir?</u>**
  1. Começando pelo áudio sem tags pre selecionado.
     a. Introduzir 1 tag.
     b. Introduzir mais 2 tags e submeter.
  2. Trocar para o áudio com tags existentes pre selecionado (2 casos: bom e mau)
     a. Pedir para classificar.
     b. Caso não utilize as sugestões, aceitar ou rejeitar as sugestões existentes.
     c. Submeter o primeiro caso deste ponto.
  3. Adicionar 5 etiquetas novas rapidamente (indicar que pode separar por virgula para inserção rápida caso necessário), das quais 2 serão negativas.
- **O que observar?**
  - Foi perceptível para o utilizador a atribuição e actualização dos pontos aquando a submissão das etiquetas? (S/N) ___

- **O que perguntar?**
  - Escolha de tags sugeridas:
    - Utilidade: (1-5) ___ Facilidade: (1-5) ___ Satisfação: (1-5) ___
  - Adicionar novas tags:
    - Utilidade: (1-5) ___ Facilidade: (1-5) ___ Satisfação: (1-5) ___
  - Possibilidade de rejeitar tags (negação):
    - Utilidade: (1-5) ___ Facilidade: (1-5) ___ Satisfação: (1-5) ___
  - Adicionar múltiplas tags rapidamente:
    - Utilidade: (1-5) ___ Facilidade: (1-5) ___ Satisfação: (1-5) ___
- Recolher opinião e sugestões do utilizador.

Tarefa 6: Utilizacão da timeline e grafo para selecionar elementos, e categorizar os mais semelhantes. (3 áudios).

- **Que pré condições são necessárias (contexto)?**
  - Segmento pre selecionado com segmentos semelhantes relevantes para a classificacao.
- **Que funcionalidades estão em causa?**
  - Possibilidade de regressar a estados e áudios anteriores com o "back" do navegador (integração da aplicação com as funcionalidades comuns encontradas nas páginas web).
- **O que avaliar?**
  - A potencialidade de retroceder em qualquer momento como funcionalidade extra na navegação entre excertos de áudio.
- **<u>O que pedir?</u>**
  1. Escutar o áudio/video actual.
  2. Escutar o áudio/video dos áudios mais semelhantes e etiquetar o audio actual.
  3. Escolher um excerto aleatório na timeline e focar/selecionar. Etiquetar o audio escolhido.
  4. Voltar ao áudio anterior (retroceder).
  5. Focar 1 dos excertos mais semelhante, recorrendo ao grafo. Etiquetar este áudio.
  6. Regressar ao áudio anterior e escolher outro excerto mais semelhante para etiquetar;
- **O que observar?**
  - Recorreu imediatamente ao botão de retrocesso do navegador web (ou respectivo atalho)? (S/N)
- **O que perguntar?**
  - Reprodução dos áudios neste contexto:
    - Utilidade: (1-5) ___ Facilidade: (1-5) ___ Satisfação: (1-5) ___
  - Reprodução dos vídeos neste contexto:
    - Utilidade: (1-5) ___ Facilidade: (1-5) ___ Satisfação: (1-5) ___
  - Utilização dos áudios semelhantes no contexto de classificação do actual:
    - Utilidade: (1-5) ___ Facilidade: (1-5) ___ Satisfação: (1-5) ___
  - Recolher opinião do utilizador.

Tarefa 7

- **O que perguntar?**
  - Que áudio iria agora classificar?
  - Considera uma boa ideia ir classificar o mais próximo (em relação a outras opções)? (1-5)
  - Em contexto de jogo, o que acha de poder ir classificar os mais proximos como modo de adquirir eficazmente mais pontos:
    - Utilidade: (1-5) ___ Facilidade: (1-5) ___ Satisfação: (1-5) ___

Tarefa 8: Categorizar 10 excertos de áudio (última tarefa do SoundsLike, penúltima tarefa geral)

- **Que pré condições são necessárias (contexto)?**
  - Nenhuma
- **Que funcionalidades estão em causa?**
  - Todas

- **O que avaliar?**
    - objectivo de analisar o ritmo do utilizador, quais os elementos mais utilizados.
    - obter uma opinião global mais precisa do utilizador
- **<u>O que pedir?</u>**
    a. Pedir ao utilizador para categorizar 10 excertos de áudio, comentando aquilo que vai fazendo ou dificuldades que encontre.
    b. Interromper ao fim de 6 áudios categorizados.

- **O que observar?**
    - Quais as secções e funcionalidades da aplicação mais utilizadas:
      Video: Ti pequena: Ti Zoom: Ti Esp: G: Tags:

    - Dispersão dos áudios no video (contiguos / dispersos).
    - Observar eventuais problemas ou dificuldades encontradas pelo utilizador:

- **O que perguntar?**
    - O facto de receber pontos pela sua correcta classificação influenciou ou compeliu a aumentar o ritmo de utilização da aplicação? (1-5) ____
    - Quanto à atribuição de pontos:
        - Utilidade: (1-5) ____ Facilidade: (1-5) ____ Satisfação: (1-5) ____

Tarefa 9: Avaliação global

- Qual a utilidade da aplicação para classificação áudio? (1-5) ___
- Qual a facilidade de utilização? (1-5) ___
- Qual a satisfação? (1-5) ___
- Voltaria a utilizar esta aplicação para classificar áudio voluntariamente? (1-5) ___
- Que frequência? _____
- Que tipo de retorno o levaria a utilizar a aplicação?
  [ ] Contribuir para classificar filmes e áudios.
  [ ] Pontos e *rankings* - competição entre pessoas e integração com redes sociais.
  Trocar pontos por serviços:
  [ ] Musica - subscrições em sites de musica como itunes, pandora ou spotify.
  [ ] Acesso a descontos em bilhetes de filmes cinema.
  [ ] Subscrições ou descontos em aluguer de vídeos online.
  [ ] Outro tipo de serviços: _____
  [ ] Outro tipo de retorno: _____
- O que mais gostou? (descritivo)

- O que menos gostou? (descritivo)

- Qual a sua opinião global da aplicação? (1-5 e descritiva se possível) ___

Escolha os termos que acha que mais se adequam para descrever a aplicação :

Compreensível	Incompreensível	Impressionante	Indefinível
Apoio	Obstructivo	Original	Banal
Simples	Complexo	Inovador	Conservador
Previsível	Imprevisível	Agradável	Desagradável
Limpo	Confuso	Bom	Mau
Confiável	Suspeito	Estético	Antiestético
Controlável	Incontrolável	Convidativo	Rejeita
Familiar	Estranho	Atractivo	Não atractivo
Interessante	Chato	Simpático	Insensível
Caro	Barato	Motivador	Desencorajador
Excitante	Aborrecido	Desejável	Indesejável
Exclusivo	Padrão		