

University of Fribourg, Switzerland



UNIVERSITÉ DE FRIBOURG SUISSE
UNIVERSITÄT FREIBURG SCHWEIZ

Joint Masters Program in Computer Science

Department of Computer Science

Guided by

Prof. Denis Lallane, Prof. Omar Abou Khalid, Prof. Elena Mugellini and
Prof. Jacques Bapst

Project Report 2011

MULTI MODAL INTERFACES

Melody Multi Modal Melody Player

STUDENTS :

Marwa Bouzeyane

Revathi priya Muthusamy

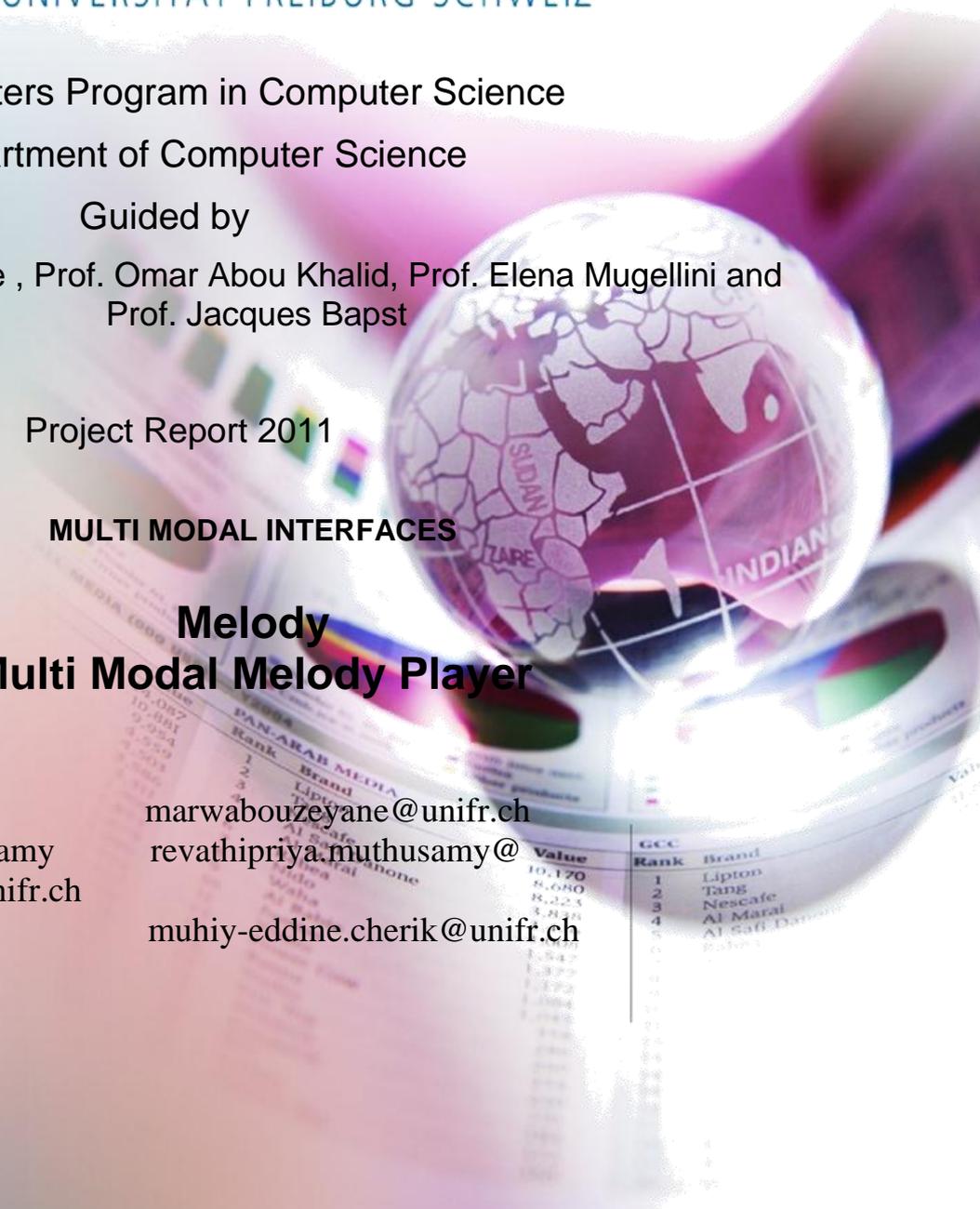
unifr.ch

Muhyiddine Cherik

marwabouzeyane@unifr.ch

revathipriya.muthusamy@

muhiy-eddine.cherik@unifr.ch



Rank	Brand	Value
1	Lipton	10,170
2	Tang	8,680
3	Nescafe	8,223
4	Al Marai	3,716
5	Al Sati-Damm	1,800
6	Balova	1,542
7		1,377
8		1,172
9		1,088
10		1,043
11		1,000
12		980
13		950
14		920
15		900
16		880
17		860
18		840
19		820
20		800

Rank	Brand	Value
1	Lipton	10,170
2	Tang	8,680
3	Nescafe	8,223
4	Al Marai	3,716
5	Al Sati-Damm	1,800
6	Balova	1,542
7		1,377
8		1,172
9		1,088
10		1,043
11		1,000
12		980
13		950
14		920
15		900
16		880
17		860
18		840
19		820
20		800

Table of Contents

1. Introduction
2. Motivation
 - 2.1.1. Tempo
 - 2.1.2. Usage
3. Development Platform
4. Architecture
5. Modalities
6. Implementation
 - 6.1.1. Installation Guide
 - 6.1.2. User Guide
7. Conclusion

1. INTRODUCTION

As part of the course "Multimodal Interfaces", we did the project to incorporate the technologies and knowledge to design, implement and evaluation multimodal user interfaces.. The application must be a kind of entertainment or serious games. Our idea is to achieve this project "MELODY". "MELODY" is an android application for easy melody reader which can play a melody file and can change the Tempo of the melody file. Selection of melody file and selection of play or pause functions are done by touch modality. The tempo of the melody which is currently playing can be changed by tapping. The volume of the melody file can dragging over the volume button. Touch and Gesture Modalities are being used in our Project.

2. MOTIVATION

2.1 Tempo :

Tempo is the speed of the rhythm of a composition. Tempo is measured according to beats per minute. A very fast tempo, prestissimo, has between 200 and 208 beats per minute, presto has 168 to 200 beats per minute, allegro has between 120 and 168 beats per minute, moderato has 108 to 120 beats per minute, andante has 76 to 108, adagio has 66 to 76, larghetto has 60 to 66, and largo, the slowest tempo, has 40 to 60.

2.2 Usage:

This tool can be used amongst DJs, concerns the speeding up or down of a record in order to match the tempo of a previous track so both can be seamlessly mixed.

DJs often beatmatch the underlying tempos of recordings, rather than their strict bpm value suggested by the kick drum, particularly when dealing with high tempo tracks. A 240 bpm track, for example, will match the beat of a 120 bpm track without slowing down or speeding up, because both will have an underlying tempo of 120 crotchets (quarter notes) per minute. Thus, some soul music (around 75-90 beats per minute) can be mixed well with a drum and bass beat (from 150-185 beats per minute).

When speeding up or slowing down a record on a turntable, the pitch and tempo of a track are linked: spin a disc 10% faster and both pitch and tempo will be 10% higher. Software processing to change the pitch without changing the tempo, or vice-versa, is called time-stretching or pitch-shifting. While it works fairly well for small adjustments ($\pm 20\%$), the result can be noisy and unmusical for larger changes.

3. DEVELOPMENT PLATFORM

3.1.1 Development Environment

- Android SDK Version 2.3
- IDE-Eclipse helios 3.6,
- Java SE 6

4. ARCHITECTURE

- Android SDK

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

Features

- **Application framework** enabling reuse and replacement of components
 - **Dalvik virtual machine** optimized for mobile devices
 - **Integrated browser** based on the open source WebKit engine
 - **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
 - **SQLite** for structured data storage
 - **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
 - **GSM Telephony** (hardware dependent)
 - **Bluetooth, EDGE, 3G, and WiFi** (hardware dependent)
 - **Camera, GPS, compass, and accelerometer** (hardware dependent)
 - **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE
- The following diagram shows the major components of the Android operating system. Each section is described in more detail below.

Android architecture

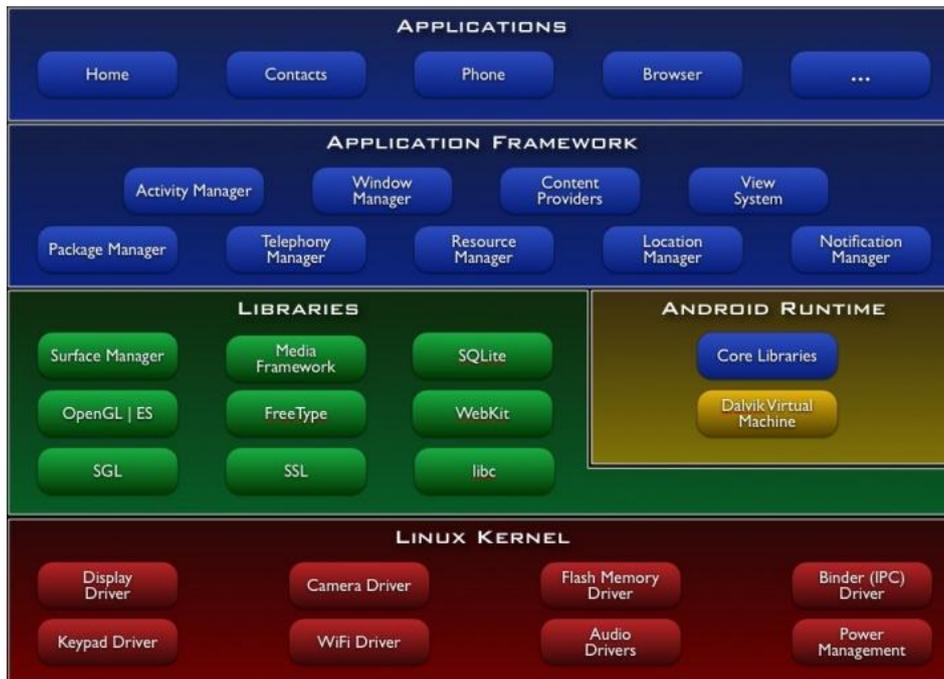


Figure 1:Android artchitectur

Gestures allow users to interact with your app by manipulating the screen objects you provide. The following table shows the core gesture set that is supported in Android.

5. MODALITIES:

Definition:

Multimodal interfaces process two or more combined user input modes (such as speech, pen, touch, manual gesture, gaze, and head and body movements) in a coordinated manner with multimedia system output. They are a new class of interfaces that aim to recognize naturally occurring forms of human language and behaviour, and which incorporate one or more recognition-based technologies (e.g. speech, pen, vision). (S. Oviatt et al., 2002).

Used Modalities:

The input methods provided in the application are:

Touch: to select and played melody

Gesture: change volume and tempo

6. IMPLEMENTATION :

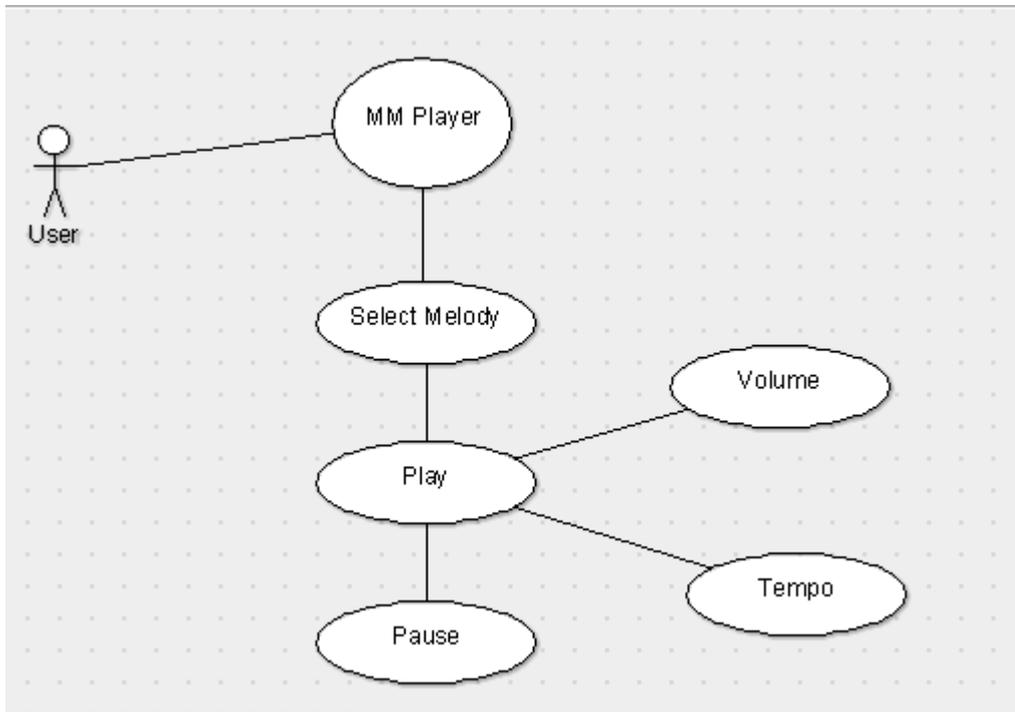


Figure 2: Use case diagram

This current GUI contains three basic melodies: In order to best manage the size of melody. It limits the user's choice to only MIDI format was created.

The diagram above represents the different functionality interface :

- select melody
- play melody
- change tempo
- Pause
- change volume

Several criteria must be present:

- Ergonomics of the GUI: The environment is easy to use, intuitive interface and design, good quality.
- Document: The documentation provided with the software quality.
- Language: The environment is in English (or French)
- Portability The software has a minimum compatibility with Windows

6.1 Installation Guide

- Install the Android SDK starter package
- Install the ADT Plugin for Eclipse.
- Add Android platforms and other packages to your SDK.
- Run the android project in Eclipse and open the result in the simulator.. Export the adk file

to the Android Smart phone using USB cable. Run the adk file by clicking the MM Player icon in the Applications menu.

6.2 User Guide

- Open the application by clicking on MM Player icon on Applications menu.

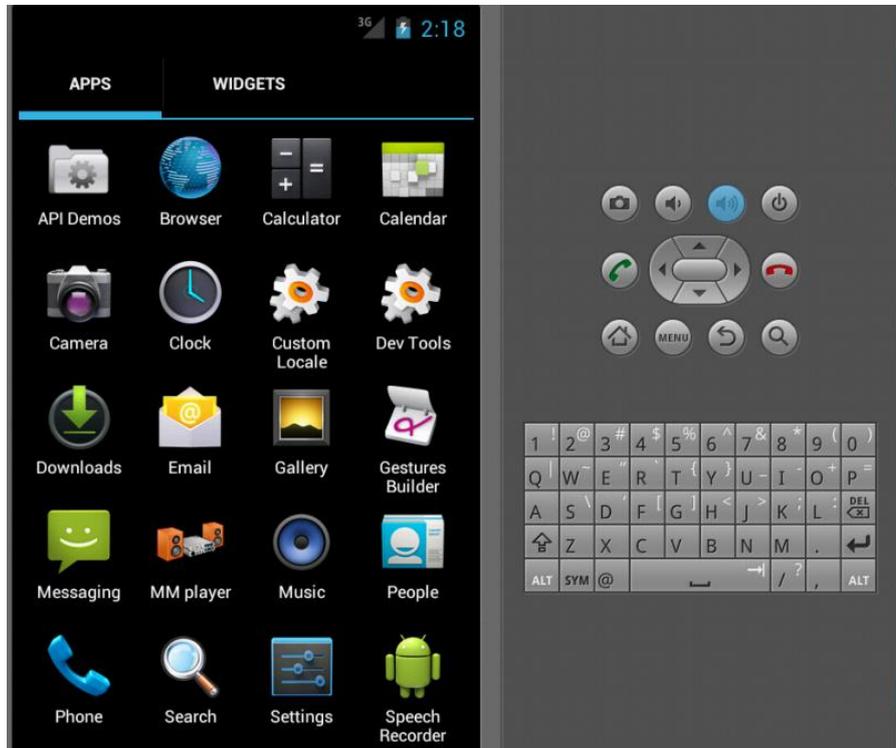


Figure 3: Home Page of MM Player

- Touch on the melody file to be played.

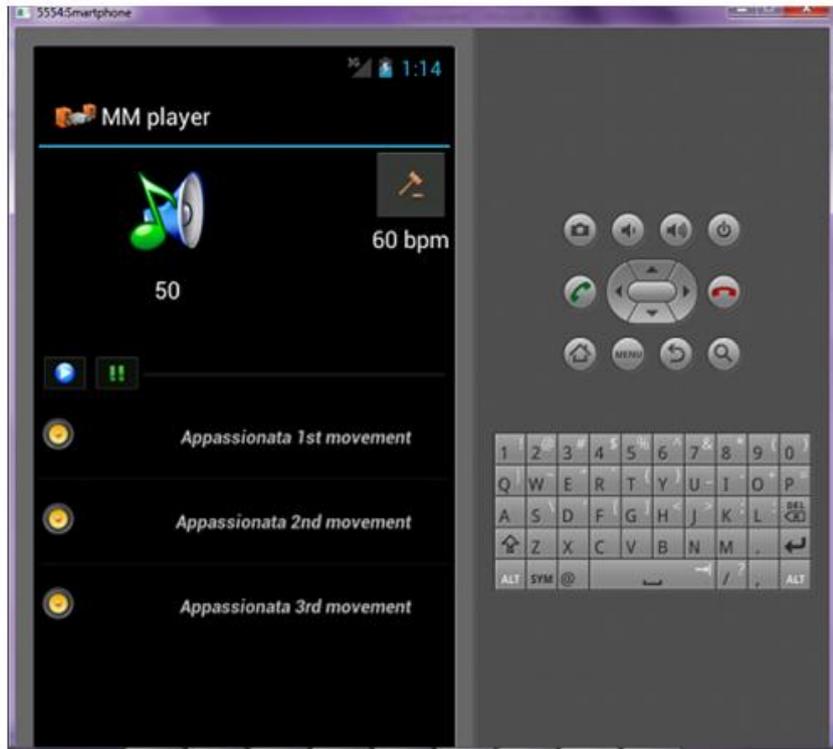


Figure 4:

Interface Melody

- Touch on the Play button to play the melody

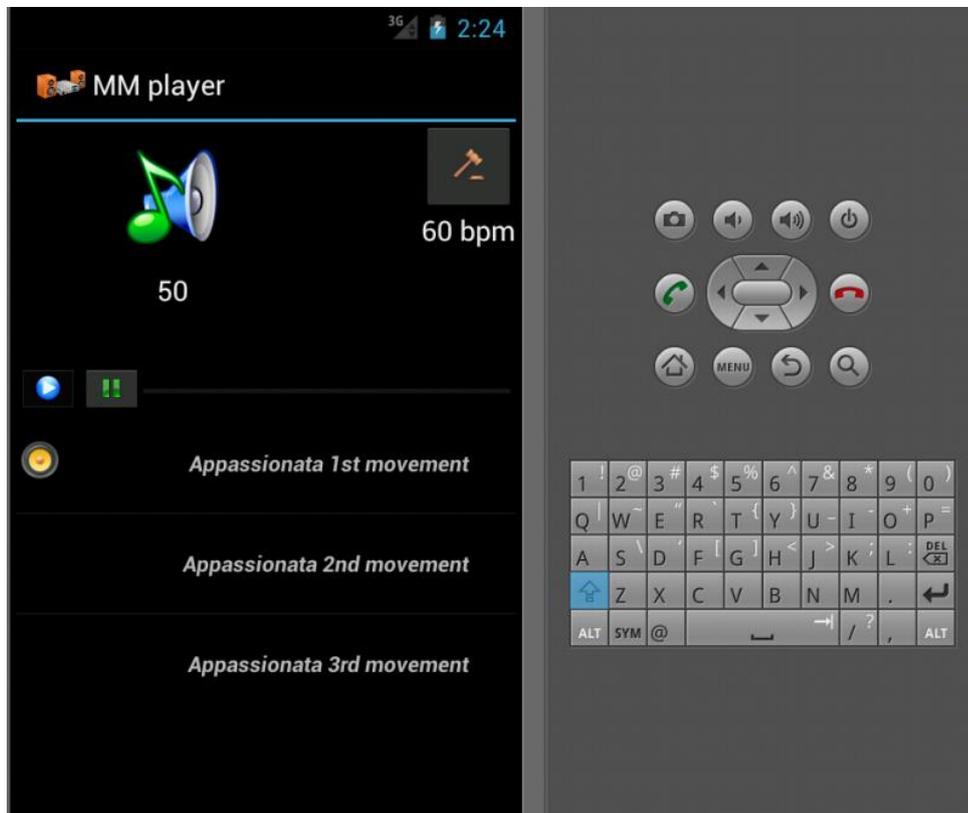


Figure 4

- Drag upwards or downwards on the volume button to increase or decrease the volume.
- Tap continuously on the BPM button to change the Tempo of the melody. The tempo can be increased or decreased.
- Touch on the pause button to pause the melody which is currently playing.
- Touch on a melody file to select the melody to be played.
- changing the volume and tempo as explained before..

7. USER EVALUATION

The method of User evaluation used for this project was the constructive interaction method, which is of qualitative nature. Though no measurable output could be gained by this, the results of the user evaluation lead to easy solutions to significantly improve the usability of the application. Besides pure usability tweaks, User Evaluation was used throughout the whole development process.

Based on user evaluation we modified the application to start with medium volume in the start. i.e the volume is set to 50 in the start of the application.

We changed the music files which are easier to detect the change in tempo for the users.

8. CONCLUSION :

With this project we gained a lot of practical skills on developing an android application with the Android SDK and Eclipse IDE. We learnt more intuitive ways of creating an application and integrating the music files with application. We achieved the goals fixed in our project proposal and are mostly satisfied with the final work what we have done. Our difficulties in this project were mainly to get familiarized with the Android SDK which was not so familiar for us before. We spent a consequent amount of time on reading books and tutorials. In conclusion, this project was very useful and good to have knowledge because it gave us the opportunity to see the workflow of developing an Android application.