

Partitures musicals. Anàlisi, disseny i prototipatge d'una aplicació mòbil Android per a reconèixer partitures musicals

Jofré Orús Aparicio

Resum— Actualment existeixen aplicacions per escanejar partitures i processar-les, però no existeixen per a dispositius mòbils. Per això, aquest projecte consisteix en, a partir d'un prototip ja existent d'aplicació Android que reconeix partitures musicals senzilles, dissenyar i prototipar una segona versió millorada que compleixi una sèrie de requeriments funcionals i no funcionals, referents tant a la interfície visual, fomentant un disseny User-Friendly i evitant ambigüitats en els fluxos de l'aplicació prototipada, com a nivell de funcionalitats, afegint noves opcions no existents a l'aplicació o millorant-ne les que ja existeixen, optimitzant-ne el rendiment o millorant les funcionalitats.

Paraules clau— Android, Optical Music Recognition, Java, disseny de Software, prototipatge, C++, Android NDK, partitures musicals, MIDI, piano, aprenentatge musical

Abstract— Nowadays, there are Applications to scan and process musical sheets, but this apps don't exist for Mobile devices. Therefore, this project consist in, from an existing Android application prototype that recognises simple musical sheets, design and prototype a second improved version that meets a list of functional and non-functional requirements relating to the visual interface design, being User-friendly and avoiding ambiguities in the flow of the application prototyping, and in terms of functionality, adding new options that don't exist in the application or improving existing ones, optimizing performance and improving functionalities.

Index Terms— Android, Optical Music Recognition, Java, Software Design, Prototyping, C++, Android NDK, musical sheets, MIDI, piano, musical learning

1 INTRODUCCIÓ

ACTUALMENT existeixen algunes aplicacions per a l'ordinador que, a partir de partitures escanejades, ens les processen i en detecten les notes, compassos, figures musicals, etc. Però per desgràcia aquest mateix tipus d'aplicacions encara no s'han implementat per a dispositius mòbils, és per això que va sorgir la idea de dissenyar i prototipar una aplicació en Android per tal de, fent una foto a una partitura, processar-la i permetre'ns escoltar-la, exportar-la en un format compatible i d'altres funcionalitats.

Per al desenvolupament d'aquest projecte hem disposat d'un prototip desenvolupat al CVC (Centre de Visió per Computador) que reconeix partitures musicals senzilles, però que consta de moltes limitacions. L'objectiu era desenvolupar una nova versió millorada d'aquest prototip, sobretot en referència a la interfície gràfica i al rendiment del prototip.

El prototip del que disposem per a començar amb el

- E-mail de contacte: jofreorus@gmail.com
- Menció realitzada: *Enginyeria del Software*.
- Treball tutoritzat per: Alicia Fornes i Fernando Vilariño (*Ciències de la Computació*)
- Curs 2015/16

desenvolupament del projecte consta bàsicament de les següents funcionalitats:

- Fer una fotografia a una partitura senzilla, i processar-la per detectar-ne les notes, compàs i armadura.
- Modificar els valors de les notes localitzades, pel que fa a la figura musical i al so.
- Escoltar la partitura.
- Tocar la partitura amb un piano.
- Guardar la partitura processada per a poder-la recuperar en el futur.
- Carregar una partitura guardada.
- Canviar els BPM (beats per minute) per escoltar-la o tocar-la.

Disposem d'una aplicació amb varies funcionalitats interessants, però amb moltes millores possibles per implementar, com per exemple aquestes:

- Eliminar notes mal detectades al processar.
- Modificar el compàs, o indicar-lo si no s'ha detectat.
- Modificar la tonalitat (armadura), o indicar-la si no s'ha detectat.
- Exportar la partitura en un format adequat, com MIDI o MusicXML.
- Millorar el processament de partitures, de forma que s'acceptin partitures més complexes.

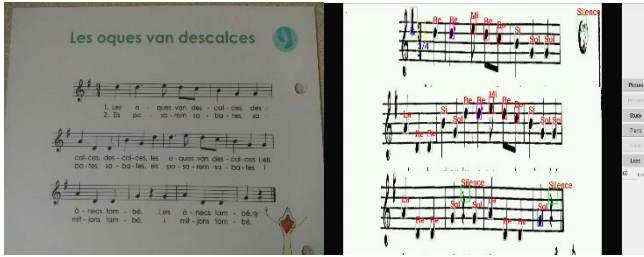


Figura 2 - Comparativa foto/partitura correcte

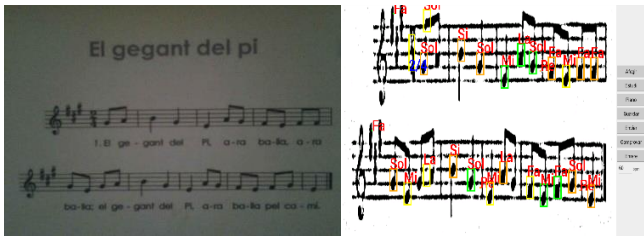


Figura 3 - Comparativa foto/partitura incorrecte

3 METODOLOGIA

Per aquest tipus de projecte, vaig considerar important utilitzar una metodologia àgil i incremental.

En aquest cas, la idea final va ser seguir alguns conceptes aplicats en la metodologia SCRUM, però centrat en el treball individual, ja que estem parlant d'una metodologia que s'aplica a grups de treball i amb molts principis col·lectius.

Així, la idea principal és realitzar entregues parcials del producte (en aquest cas la nostra aplicació), prioritant el benefici que n'obtindrà el receptor. La metodologia és adient per un tipus de projecte així, ja que fàcilment els requeriments poden canviar en el transcurs del projecte, i d'aquesta manera els podem modificar després de cada sprint de programació, apart ens serveix per fomentar la innovació i per assimilar en la millor mesura que sigui possible el codi original.

El punt clau que s'ha utilitzat de la metodologia SCRUM és el desenvolupament en sprints, de forma que en cada sprint de programació ens proposàvem uns requeriments a implementar i després es valorava amb els tutors si la feina feta era la adient i s'analitzaven els següents requeriments a implementar.

Durant el transcurs del projecte hi ha hagut 3 sprints, on s'han assolit els següents objectius:

- 1r sprint: Implementació del sistema de modificació, inserció i eliminació dels elements de la partitura, multi-idioma i correcció d'errors detectats.
- 2n sprint: Processar arxius d'imatge, exportar a MIDI, comprovar consistència de la partitura, donar noves opcions al piano i correcció de nous errors detectats.
- 3r sprint: Afegir puntuació al tocar el piano, enviar

Mail amb l'arxiu MIDI generat, canvi de disseny i correcció d'errors nous detectats.

A l'apartat de desenvolupament estan explicats més detalladament els objectius assolits a nivell de programació.

4 PLANIFICACIÓ

La planificació inicial del projecte constava de les següents etapes:

- Etapa 1: Primer contacte amb el codi + Anàlisi de requeriments
 - Del 21/09/15 al 11/10/15
- Etapa 2: Disseny de l'aplicació + Desenvolupament de l'aplicació (1r sprint)
 - Del 12/10/15 al 15/11/15
- Etapa 3: Desenvolupament de l'aplicació (2n sprint) + Validació de la feina feta
 - Del 16/11/15 al 20/12/15
- Etapa 4: Validació de la feina feta + Revisió de requeriments i disseny + Últimes tasques de desenvolupament (3r sprint)
 - Del 21/12/15 al 22/01/16
- Etapa 5: Entrega final i presentació del projecte
 - Del 25/01/16 al 15/02/16

La planificació es va seguir correctament en les dues primeres etapes, però a partir de la 3a va sofrir modificacions provocades pel l'augment de feina i dificultat experimentat. D'aquesta manera la validació que havia de començar a l'etapa 3 finalment es va començar a la següent etapa de la planificació, apart aquesta es va allargar una setmana més per tal d'assolir correctament els objectius. D'aquesta manera a partir de l'etapa la planificació va canviar d'aquesta manera:

- Etapa 3: Desenvolupament de l'aplicació (2n sprint)
 - Del 16/11/15 al 17/01/16
- Etapa 4: Fase de validació de les proves + Últimes tasques de desenvolupament (3r sprint)
 - Del 18/01/16 al 29/01/16
- Etapa 5: Entrega final i presentació del projecte
 - Del 30/01/16 al 15/02/16

Finalment, la revisió de requeriments i disseny posterior a la fase de validació no ha sigut tant extensa com s'havia volgut per falta de temps, però tot i així s'han pogut revisar petites implementacions del disseny i els requeriments gràcies a les proves fetes durant la fase de validació. A més hi ha una extensa valoració sobre aquestes proves que ens ha servit per indicar possibles millores a desenvolupar en el futur respecte aquest projecte en concret.

5 DESENVOLUPAMENT

En aquest apartat explicaré les fases de desenvolupament realitzades durant el transcurs d'aquest projecte.

5.1 Anàlisi de requeriments

Abans de començar amb la implementació del projecte, es va dur a terme un anàlisi previ de requeriments necessaris per a complir els objectius del projecte. Al utilitzar-se una metodologia àgil, durant el transcurs del projecte (després dels sprints) van aparèixer nous requeriments, però els que exposaré a continuació són els principals que es van plantejar.

Aquesta és la llista de requeriments funcionals:

- Afegir notes no detectades per l'aplicació.
- Eliminar notes detectades per l'aplicació o afegides per l'usuari.
- Aplicar els canvis fets en l'edició de la partitura a l'hora d'escoltar-la o tocar-la.
- Modificar el so de les notes en una octava més.
- Afegir, canviar o eliminar l'armadura.
- Afegir, canviar o eliminar el compàs.
- Modificar la figura de les notes en més opcions de les disponibles.
- Eliminar partitures processades i guardades.
- Processar arxius d'imatge de la Tablet.
- Exportar un arxiu MIDI i/o Music XML.

I a continuació la llista de requeriments no funcionals:

- Canviar l'UI per una més usable i intuïtiva (usabilitat).
- Canviar els desplegable per editar les notes musicals (usabilitat).
- Afegir i canviar colors de les figures musicals, i afegir una llegenda (usabilitat).
- Activar/desactivar botons de la UNI quan no es puguin utilitzar (fiabilitat).
- Processar la partitura amb una efectivitat més alta a l'hora de trobar les notes (rendiment).

5.2 Interfície gràfica

Per començar amb la millora de la interfície gràfica, vaig dissenyar un mockup basat en el prototip del que disposava però amb les noves funcionalitats que sorgien de l'anàlisi de requeriments. Aquest mockup es va fer amb l'eina online Moqups que permet fer el disseny bàsic i apart enllaçar esdeveniments per simular el funcionament de l'aplicació.

A continuació mostro una sèrie de pantalles del mockup.

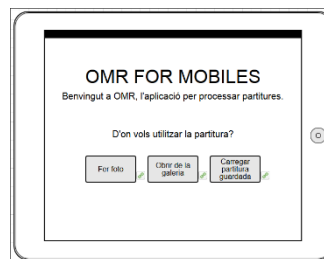


Figura 4 - Mockup pantalla principal

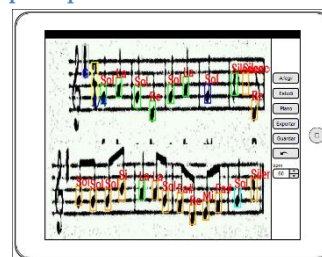


Figura 5 - Mockup partitura processada

La idea de l'aplicació a nivell de disseny (sobretot al tractar-se d'un prototip), consistia en tenir una única pantalla principal que al principi ens permetés carregar la partitura des de diferents formats (fotografia, arxiu o partitura guardada des de l'aplicació), i un cop tinguéssim la partitura processada des de la mateixa activitat donar-nos les opcions indicades en els requeriments (afegir elements, tocar el piano, escoltar la partitura, guardar-la...).

Finalment algun detall dissenyat al mockup no és exactament com es va pensar llavors, ja que durant el desenvolupament han sorgit nous requeriments que han fer modificar el disseny de la interfície gràfica, aquí teniu un exemple de com queda finalment l'aplicació un cop la partitura ja s'ha processat:



Figura 6 - App després de processar la partitura

A nivell d'implementació, l'estructura de classes feta servir segueix el model utilitzat en el prototip proporcionat pel CVC, a continuació exposaré les classes utilitzades per al desenvolupament del projecte i a quina funcionalitat van destinades:

- Pantalla principal
 - MainActivity.java
- Càmera
 - Preview.java
 - AutofocusCamera.java
- Fer foto
 - PhotoHandler.java
 - PictureProcess.java
- Afegir/modificar/eliminar elements
 - TouchImageView.java
 - KeyDialog.java
 - TimeDialog.java
 - StudyDialog.java
- Escoltar la partitura
 - PlayMusic.java
 - MidiDriver.java
- Exportar a MIDI
 - MidiExport.java
- Enviar MIDI per Mail
 - GMailSender.java
 - JSSEProvider.java

Apart de la pantalla principal, cada cop que una de les funcionalitats implementades necessita de la interacció de l'usuari necessitem de l'ús de diàlegs on l'usuari indica la informació necessària, com en els casos de modificació d'elements de la partitura, guardar, comprovar la consistència de la partitura i d'altres. Alguns d'aquests diàlegs, com per exemple el de modificació de notes, generaven dubtes en el seu funcionament, i s'han hagut de modificar per tal que l'usuari entengui ràpidament la informació que se li demana, sense ajuda externa.

A més, s'han afegit varis idiomes a l'aplicació ja que el prototip disposava d'algunes paraules en anglès i altres en català. A partir de varis arxius strings.xml s'ha adaptat l'aplicació per als idiomes català, castellà i anglès, depenent de l'idioma que tinguem a la nostra Tablet.

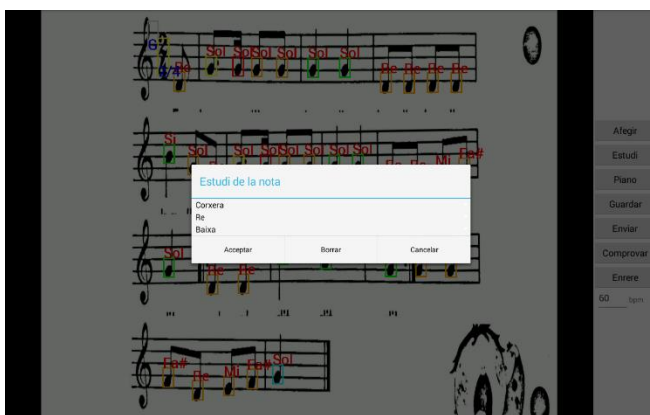


Figura 7 – Estudi de la nota modificat

4.2 Processament de les imatges

Aquesta part ja estava implementada en el prototip original, i finalment per falta de temps i de prioritat no s'ha modificat. Tot i així és interessant analitzar-ne el

funcionament per entendre com ens ho farem després per modificar la partitura i aplicar les funcions necessàries que havíem pensat.

Per processar la partitura, l'aplicació agafa la imatge proporcionada (ja sigui a partir d'una foto o un arxiu) i, utilitzant funcions de la llibreria OpenCV de C++ (amb Android NDK) crea una nova imatge corregint la inclinació de la imatge original, cerca els pentagrames de les partitures, intenta indicar-ne el compàs i l'armadura i cerca les notes de cada pentagrama, indicant-ne també la figura i el to. Al final fa una comprovació de la consistència de la partitura, per tal d'intentar corregir automàticament alguna nota que no s'hagi detectat correctament.

Finalment, se'ns retorna la nova imatge i en aquesta se'ns dibuixen caixes per el compàs, l'armadura i les notes detectades. Les notes se'ns guarden en una classe que hem creat anomenada "Box" que conté la informació de les mides de la caixa, la posició en la imatge, la duració de la nota, el to i la figura musical. Aquesta classe és important ja que després a partir d'una ArrayList<Box> tindrem accés a totes les notes i podrem escoltar la partitura, modificar-ne les notes o eliminar-ne, tocar el piano o comprovar la consistència de la partitura.

4.3 Afegir, modificar i/o eliminar elements

El prototip original donava l'opció de, tocant les notes de la partitura processada, donar-los un altre valor en referència al seu to (Do, Re, Mi...) i a la seva figura (negra, silenci de corxera, blanca...). Tot i així aquest sistema constava d'algunes limitacions per corregir la partitura:

- El to de la nota a modificar només es podia canviar en l'octava corresponent a la nota (no podíem canviar un Mi4 per un Mi3, però sí per un Re4)
- No es podien eliminar ni afegir notes.
- No es podia ni afegir, ni modificar ni eliminar el compàs
- No es podia ni afegir, ni modificar ni eliminar l'armadura (tonalitat).

Això provocava que, a no ser que obtinguéssim un processament de la partitura pràcticament perfecte, no poguéssim corregir tots els errors trobats.

La implementació més interessant d'aquesta part és la d'inserció de noves notes no detectades. L'estructura de dades de les notes, com he dit abans, es correspon a un ArrayList<Box> amb les notes ordenades, per això quan afegim una nota, segons la seva posició a la imatge hem de determinar la seva posició en l'ArrayList. Per fer-ho, utilitzarem el següent algoritme:

```

// Afegim la nova nota
Box newBox = new Box(viewX - xMean, viewX + xMean,
viewY + yMean, viewY - yMean, 5, 60, 750);
// Busquem la nota abans de la que acabem d'afegir
per situar-la
while (i < boxes.size()) {
    x = boxes.get(i).leftx;
    y = boxes.get(i).topy;
    // Calculem la distància euclidiana entre la
nova nota i la nota que estem recorrent
    distance = Math.sqrt(Math.pow(viewX - x, 2) +
Math.pow(viewY - y, 2));
    // Si es el primer cop que entrem al loop
guardem la distància com la més propera
    if (i == 0) {
        nearDistance = distance;
    } else {
        // Sinó, mirem si està més aprop que abans
i que la nota que recorrem no estigui més endavant
en la partitura
        if (distance < nearDistance && viewX > x)
        {
            nearDistance = distance;
            nearI = i;
        }
    }
    i++;
}
// Afegim la nota a la seva posició corresponent i
tornem a dibuixar les notes
boxes.add(nearI + 1, newBox);

```

D'aquesta manera, la nota queda en la seva posició real dins de l'ArrayList i al escoltar o tocar la partitura funciona correctament.

Una altra implementació interessant d'aquest apartat és la modificació de l'armadura, ja que aquesta modificació pot suposar que algunes notes de la partitura canviïn de to, per exemple, si d'una partitura en DoM (cap alteració) la canviem a SolM (un sostingut) totes les notes Fa de la partitura hauran de passar a ser un Fa#.

Per fer-ho, segons l'armadura analitzem totes les notes de la partitura i quan trobem un valor conflictiu li canviem el to.

4.4 Escoltar la cançó

Aquesta funcionalitat també estava implementada, només tenia un error, al modificar una nota no li canviava la duració, de manera que al escoltar la partitura no s'identificava aquest tipus de modificació.

Per tant, referent a aquesta funcionalitat del prototip només ha fet falta actualitzar la duració de les notes (classe Box) al modificar-les, a més un error de programació feia que els BPM (beats per minute) fossin més alts dels que realment indicàvem al reproduir el tema, problema que també vaig resoldre correctament.

Aquesta funcionalitat, apart de l'atractiu de poder escoltar la partitura que teníem en paper, també ens pot servir per detectar-ne errors si ja la coneixíem, per tant és una eina important pels usuaris que no tinguin coneixements musicals per veure si tot està bé.

4.5 Tocar la cançó

Aquesta és, juntament amb tot el referent a la correcció de la partitura, la part més treballada i interessant del projecte.

Originalment, l'aplicació permetia tocar la cançó que haguéssim processat, però hi havia limitacions destacades en el desenvolupament d'aquesta part:

- El piano era molt petit, era complicat de tocar correctament.
- No se'ns indicava si ens equivocàvem de nota.
- La nota sonava al deixar anar el dit de la tecla, donant una sensació de poc realisme.
- El refresc de la partitura (on se'ns indicava la nota que havíem de tocar) tenia un cert retard.

Apart, la opció de tocar el piano només estava disponible seguint al peu de la lletra la partitura i amb ajuda (se'ns indicava la nota a tocar al piano). Per aquest motiu durant el transcurs del projecte van sorgir nous requeriments a desenvolupar sobre aquesta part:

- Tocar el piano seguint la partitura sense indicar la nota a tocar.
- Tocar el piano lliurement, sense seguir la partitura.

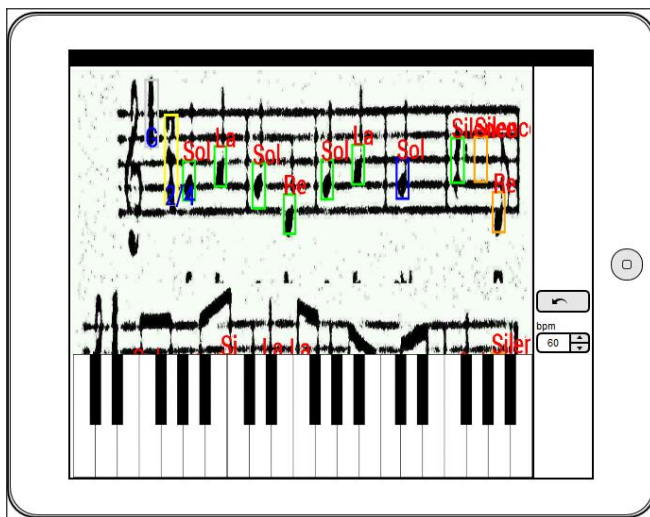


Figura 8 - Mockup piano modificat

Totes aquestes limitacions es van solucionar a la classe principal de l'aplicació (MainActivity.java), aplicant les següents mesures:

- Ampliar la mida del piano horitzontalment, per adaptar-lo al màxim de l'amplada de la Tablet.
- Fer un soroll d'error al equivocar-nos de nota.
- Canviar la programació interna perquè la nota sonés quan cliquéssim la tecla, i es renovés la imatge al deixar anar el dit.
- Optimitzar la funció per tal d'evitar el retard en el refresc.
- Afegir un diàleg amb les opcions disponibles per

tocar el piano al iniciar l'activitat.

- Sense ajuda: La imatge del piano no canvia en tot el procés.
- Estil lliure: Determinem la tecla que hem tocat a partir de la posició on cliquem.

A més, a l'últim moment, i pensant en fer més entretinguda aquesta opció del prototip (fomentant l'aprenentatge musical, però d'una forma divertida), es va afegir una puntuació final al acabar de tocar, per tal de generar una mica de competitivitat entre els usuaris finals. La fórmula de càlcul de la puntuació és aquesta, la puntuació màxima és 100 i la mínima és 0:

- $(\text{notesEncertades} * 100 / \text{nombreNotes}) - (\text{notesFallades} * 100 / \text{nombreNotes})$

4.6 Guardar la partitura

Aquesta opció ja estava implementada, només es van solucionar alguns errors lleus (sobreescriure arxius guardats si no era el primer cop que guardàvem la partitura, per exemple), i es va afegir dins d'aquesta funcionalitat l'exportació de la cançó a MIDI en els BPM indicats per l'usuari en el moment de guardar, en la següent subsecció explicaré aquest apartat.

Només indicar que el guardat de les partitures està comprès per 3 arxius:

- `img.png`: La imatge processada de la partitura.
- `imgOrig.png`: La imatge original.
- `boxes.xml`: XML amb la informació de les notes (classe `Box`) de la partitura, per tal de tornar-les a dibuixar al carregar una partitura guardada.

El format en que es guarden les notes a `boxes.xml` és el següent:

```
<Box id="0" name="Sol" bottom="411.4286" top="275.39172" left="368.39792" right="413.02344" frequency="67.0" duration="1000.0" type="5"/>
```

4.7 Exportar a MIDI

Arribats a aquest punt, és interessant explicar que és l'estàndard MIDI.

MIDI (Musical Instrument Digital Interface) permet que varis instruments musicals electrònics, ordinadors o altres dispositius relacionats es connectin i comuniquin entre si. El sistema MIDI transmet missatges d'esdeveniments que especifiquen la notació musical, el to i la velocitat, senyals de control per paràmetres musicals com són la dinàmica, el vibrato i d'altres, i senyals de rellogte per sincronitzar el tempo entre varis dispositius.

Al final el que ens interessa a nosaltres de l'estàndard

MIDI és poder guardar una melodia consistent en les notes de la partitura en un tempo concret, per a fer-ho hem utilitzat una llibreria que ens permet generar arxius MIDI donant la informació de les notes de la partitura, el tempo i el compàs.

Per a desenvolupar aquesta part, hem utilitzat una llibreria d'Android anomenada "Android MIDI Library", que entre d'altres ens permet crear arxius MIDI. Per això hem creat una classe intermitja anomenada "MidiExport" que ens serveix de connexió entre les nostres classes originals i les de la llibreria especificada, convertint les notes (classe `Box`) que tenim a informació per escriure un arxiu MIDI.

4.8 Comprovar la consistència de la partitura

Aquesta és una utilitat per als usuaris finals molt senzilla, però que els pot servir, sobretot sinó tenen gaire nivell d'aprenentatge musical, per a comprovar que la consistència de la partitura a nivell rítmic sigui la correcta.

Per això, demanem a l'usuari el nombre de compassos que té la partitura (aquest no es calcula en el processament de les partitures), i també necessitem que el compàs estigui especificat.

A partir d'aquí, un senzill càlcul amb la duració de totes les notes i la quantitat de compassos multiplicada per la mesura del compàs ens serveix per indicar si la consistència rítmica de la partitura es compleix o no.

4.9 Enviar mail amb la partitura

En l'última etapa de desenvolupament del projecte, abans de passar la fase de validació, es va plantejar l'opció de que els usuaris que provessin l'aplicació poguessin emportar-se la melodia de la cançó processada com a record de l'experiència.

Per a fer-ho, vaig utilitzar la `JavaMail API`, que a partir d'uns mètodes concrets ens permet enviar un Mail des de una compte nostra a qui vulguem. L'única modificació de codi que es va haver de fer va ser afegir l'opció d'adjuntar fitxers als Mails enviats.

Finalment, després de la fase de validació es va valorar que era interessant afegir la partitura processada al Mail enviat, de manera que els usuaris poguessin rebre un complement més al enviar-se el Mail.

5 VALIDACIÓ

Un cop acabat el desenvolupament del projecte, es va realitzar la fase de validació per tal de comprovar si els usuaris finals valoraven positivament els canvis fets en l'aplicació i també per detectar-ne mancances o errors.

Apart, durant el transcurs del projecte també s'han realitzat una sèrie de proves per comprovar que les funcionalitats de l'aplicació seguien funcionant correctament després de cada sprint, com explicaré a continuació.

5.1 Test

Durant els sprints de programació duts a terme durant el projecte es van fer, al finalitzar cada un d'aquests, proves de regressió per comprovar que l'aplicació seguia funcionant correctament.

Aquestes proves no estan documentades, però es basen en seguir els diferents camins del diagrama de flux de l'aplicació per tal de comprovar totes les opcions possibles i veure si el comportament és l'adequat. Tots els errors trobats durant aquesta fase es convertien en tasques prioritàries a solucionar en el següent sprint de programació.

5.2 Validació amb usuaris finals

La fase de validació amb usuaris finals es va desenvolupar al Library Living Lab de Volpelleres, un espai dins d'una biblioteca dedicat a ensenyar als usuaris aplicacions que els puguin servir per a diferents aprenentatges, en el nostre cas l'aprenentatge musical.

Les proves es van realitzar durant una setmana, en la qual es mostraven diferents aplicacions per fomentar l'aprenentatge musical, entre elles la meua. La idea era que els usuaris provessin l'aplicació des del principi del seu funcionament fins al final d'aquest, descobrint totes les funcionalitats que aquesta tenia.

Les instruccions que seguien els usuaris eren les següents:

- Fer la foto i processar la imatge.
- Corregir les notes detectades incorrectament, afegir les que faltin.
- Escoltar la cançó.
- Tocar el piano, en les opcions que es vulgui.
- Guardar la partitura i enviar-la per Mail juntament amb l'àudio.

Per tal que els usuaris poguessin seguir els passos i al mateix temps valorar-los positiva o negativament, se'ls entregava un formulari amb les instruccions a seguir i un requadre per a cada una per valorar-la amb diferents "cares" (5 expressions de més trist a més trist) de for-

ma positiva o negativa.

A més dels formularis, el "feedback" amb els usuaris va ser essencial per comprovar quines funcionalitats eren més intuïtives i quines altres s'havien d'explicar més o no s'entien, també va servir per trobar errors i/o noves idees per l'aplicació que, al haver de tancar el projecte, no s'han pogut desenvolupar, però si tenir-les en compte en el futur.

Els formularis també preguntaven l'edat dels usuaris i si tenien coneixements musicals, per tal de conèixer millor les impressions dels usuaris per sectors concrets.

OMR – Aplicació per a l'aprenentatge musical

Full de recollida de dades sobre l'experiència d'ús.

QUINA EDAT TENS? 10

TENS CONEIXEMENTS MUSICALS? No

1
FÉ UNA FOTO A LA PARTITURA QUE PREFEREIXIS, I APRETA EL BOTÓ "PROCESSAR"

2
CORREGIEX LES NOTES I/O ELEMENTS INCORRECTES DE LA PARTITURA CLICANT-HI A SOBRE O AMB EL BOTÓ "AFEGIR"

3
ESCOLTA LA CANÇÓ AMB EL BOTÓ "ESTUDI", COMPROVA QUE SONI CORRECTAMENT

4
TOCA LA CANÇÓ AMB EL PIANO, INTENTA FER LA MILLOR PUNTUACIÓ POSSIBLE EN LES DIFERENTS MODALITATS

5
PER ÚLTIM, GUARDA LA PARTITURA I ET PODRÀS EMPORTAR LA CANÇÓ CAP A CASA

Library Living Lab
Biblioteca Pública Miquel Barberí
Sant Cugat del Valles - Barcelona

member of
European
Network of
Living Labs

Really good

Really good

Brilliant

Brilliant

Brilliant

Figura 9 - Formulari que s'entregava al Living Lab per valorar l'aplicació

Tot i que finalment no van venir suficients usuaris per tal de dur a terme un bon estudi, els resultats que es van obtenir són positius, sobretot en la franja d'edat de 6 a 12 anys, que majoritàriament són els usuaris que van passar a fer les proves i a més són els més indicats per utilitzar l'aplicació.

Els formularis entregats pels usuaris ens van permetre extreure informació sobre cada funcionalitat en concret. Per exemple, els usuaris van valorar molt positivament poder escoltar i tocar la partitura amb diferents modalitats de piano, mentre que durant el transcurs de les proves es va comprovar que la funcionalitat d'enviar per

Mail l'àudio no funcionava correctament, i per tant es va valorar negativament als formularis.

Després de la fase de validació es van solucionar una sèrie d'errors detectats durant aquesta fase:

- Els adjunts del Mail no s'obrien correctament.
- Canviar el nom del "silenci" per "silenci de negra".
- Revisar el funcionament del canvi de tonalitats.
- Revisar el multi idioma.
- Al afegir una nota al principi, es posa en segona posició.
- Si no tanquem la partitura i la tornem a carregar no la podem enviar per Mail.

Finalment, per tal de que els usuaris poguessin resoldre els seus dubtes de funcionament, es va desenvolupar un manual d'usuari.

7 CONCLUSIÓ

Un cop acabat el desenvolupament d'aquest projecte, els resultats són majoritàriament satisfactoris i positius. Hem passat de tenir un prototip que, a no ser que procesés una fotografia perfecte, no ens donava l'opció de deixar-la exactament igual a l'original, a tenir un nou prototip millorat que ens permet corregir la partitura si el processament no és del tot satisfactori, tocar-la de diferents formes amb un piano, exportar-la a MIDI i enviar-la per Mail, entre d'altres.

Tot i així no totes les conclusions són positives, una part de l'aplicació que hagués sigut interessant millorar és la referent a la visió per computador, o sigui al processament en si de la partitura. La falta de coneixements per part meua en aquest sentit han propiciat que, després del primer mes de treball, es descartés aquesta opció, tot i així, crec que per tancar el projecte del tot i passar de tenir un prototip a una aplicació totalment funcional és imprescindible millorar-la considerablement en aquest sentit, i aquesta tasca seria molt interessant que fos duta a terme pròximament en el TFG d'algun alumne de la carrera.

Finalment, dir que a nivell personal estic content de la feina feta, sobretot pels coneixements apresos a nivell d'Android durant el projecte, i també per la validació dels coneixements que he anat aprenent durant aquests anys de carrera, considero que aquest projecte a simulat correctament el desenvolupament d'un software en les seves etapes i que em pot servir com a experiència en el meu futur professional.

AGRAÏMENTS

A la meua família, sobretot als meus pares, perquè si no fos per ells no crec que hagués estudiat ni informàtica ni música.

Als meus amics, en especial als "Cyclers" i als "Esparidenyes", ells ja saben qui són.

Als components dels meus grups de música, "L'Òstia del Segle" i "Eskassa Llibertat" per tots els dies que no he pogut anar a assajar per estar treballant en aquest projecte o altres coses de la universitat.

Als meus tutors de TFG, pels coneixements aportats i l'ajuda que m'han donat durant aquest projecte.

BIBLIOGRAFIA

- [1] <https://github.com/leffelmania/android-midi-lib>
- [2] <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Rational+Team+Concert+for+Scrum+Projects/page/SCRUM+como+metodolog%C3%ADa>
- [3] <http://l3.cvc.uab.es/>
- [4] <https://en.wikipedia.org/wiki/MIDI>
- [5] <http://developer.android.com/intl/es/ndk/index.html>
- [6] <http://tools.android.com/tech-docs/android-ndk-preview>
- [7] <http://www.musoft-builders.nl/music/accessing-midi-in-delphi-forandroid/>
- [8] <http://developer.android.com/intl/es/index.html>
- [9] <http://stackoverflow.com/questions/2020088/sending-email-in-android-using-javamail-api-without-using-the-default-built-in-a>
- [10] <http://www.cugat.cat/docs/Volpelleres-Living-Lab.pdf>
- [11] <https://moqups.com/jeff039/ueBH74o3/>

APÈNDIX

A1. DIAGRAMA DE FLUX DE L'APLICACIÓ

