Training Materials and Course Design for The Fundamentals of Music

Submitted by Joshua Veltman,
Professor of Music History & Literature,
in consideration for the Newell Innovative Teaching Award, 2015
Newell Innovative Teaching Award Application, 2015
Training Materials and Course Design for The Fundamentals of Music

A. Description

1. Background. In the fall semester of 2014, I was tasked with teaching a new class in the music curriculum, The Fundamentals of Music. This 8-week accelerated music theory class caters to students whose backgrounds had not prepared them to begin the music theory sequence with Elementary Music Theory or who had struggled in the Elementary course. The visual identification and writing of basic musical structures such as scales, intervals, and triads formed the main subject matter of this course. In a remedial setting such as this, I determined that the students needed a great deal of hands-on practice and needed to focus on step-by-step skill building. Some kind of computerized training was called for. Conveniently, the class was scheduled to meet in the music technology lab, with computers readily available. Music theory websites and software programs already exist, but none seemed to suit the particular circumstances of this course. Pre-existing packages tend to focus on ear training (aural identification of musical structures), which, while important, was not the focus of the Fundamentals course. Pre-existing packages also tend to be comprehensive in scope (and expensive), offering far more than what the Fundamentals students would need. So, I decided to create my own customized, cost-effective solution. I developed a series of online drills to be delivered through MoodleRooms, Union’s current LMS (Learning Management System).

2. Online Drills. The most effective online drills would feature a randomized selection from all possible permutations of a given musical structure. (A small subset of examples could simply be memorized and would not require students to actually understand the theory behind the correct answers.) This called for a vast array of examples (typically in the hundreds or thousands, depending on the structure), but it would have been time-prohibitive to construct these individually. I turned to computer coding to solve the problem. I used a combination of shell script (a Unix-based programming language), MUP (music publishing software that accepts plain-text input), and XML (extensible markup language, for importing questions into the MoodleRooms environment). With these tools I created drills that included graphics of the musical examples as well as supporting audio.

For an example of one of the many scripts I wrote to generate the music graphics and audio for an online drill, please see Appendix I. Appendix II shows part of a script to generate the XML code that creates drill questions in MoodleRooms. For a short demonstration of an online drill, please point your web browser to: http://museprof.net/newell/demo.html
3. Course Design. The particular circumstances of the course called for a rethinking of the traditional music theory classroom experience. The design of the new course, built around the online drills, features step-by-step skill building, frequent, low-stakes assessment with immediate feedback, and cumulative review at every step. A typical class session starts with a short quiz to assess knowledge and skills related to the previous lesson. Each quiz contains a few questions of cumulative mixed review. Quizzes are graded in class by peers for immediate feedback. Then I teach a short lesson to introduce a new concept, after which students log on to the lab computers and begin working on drills that reinforce the new concept. I circulate among the students to answer questions and provide support. Between classes, for homework, students will continue working on the drills and will hopefully spend as much time as necessary to achieve proficiency. Each series of drills ends with a mixed review that reinforces all concepts and skills previously learned.

The drills for a given topic are carefully sequenced so as to isolate component skills and then add them together, progressing from easier to more difficult. In order to be deemed proficient, a student must complete a given drill at a 90% accuracy rate within a given time limit. Students may redo a drill as many times as needed, and they must “pass” a drill before the system allows them to move on to the next one. For each topic, students are awarded a Drill Completion Grade, receiving an A for all drills completed with proficiency, a B for all but one drill completed, a C for all but two, and so on. This procedure introduces an element of “gamification” to the course insofar as students must “master a level” before moving on to the next one, and are incentivized with rewards (grades) accordingly.

B. How the project differs from current teaching methods

Typical music theory courses are centered around paper-and-pencil homework that is non-repeatable and has delayed feedback. If computerized exercises are used, they are usually supplemental and done outside the classroom. Assessments in music theory courses typically consist of a few high-stakes items (a few tests and a final exam). Some students succeed under this model, but there is always a subset of students that seems to struggle with it.

These are the key differences of my approach:

- online drills with immediate feedback are an integral part of the daily classroom activity
- online drills focus on step-by-step skill building, and students are rewarded with grades for successfully completing each step
- the online drills are tailored specifically for this class and have not been published elsewhere
- the daily quizzes are frequent, low-stakes assessments with immediate feedback
• the course is tailor-made for a particular population of students who have no background in music theory or who have struggled in the entry-level theory course; Fall 2014 was the first time such a course has been offered at Union

C. Evaluation

The inaugural run of The Fundamentals of Music was definitely a success, especially considering the various challenges it faced (new course in the curriculum, an instructor who was teaching outside his main area, and an underprepared student population). The students quickly adjusted to the design and expectations of the course. They also expressed appreciation for the custom-built online drills that allowed them to work at their own pace and reinforce their skills with an infinitely patient computerized “tutor”. All students passed the class, with a majority earning A’s and B’s. This allowed them to move confidently into the next step in the music theory sequence.

There are still a few “bugs” in the system that need to be squashed, such as correct answers being marked incorrect. I would also like to even out the number of drills per topic, as some topics have a long series of drills and others have only a few. This will require careful thinking about what steps could be combined or split apart. I plan to redo the earlier drills to include an audio component, since I learned how to generate the audio partway through the course. I would like to increase the “gamification” of the drills, and before too long I hope to acquire the programming skills needed to offer them on mobile devices like tablets and iPhones.
Appendix I: Example of a script to generate the music graphics and audio for an online drill

rm -r jpg
rm -r mp3
mkdir mup
mkdir ps
mkdir jpg
mkdir mupformid
mkdir mid
mkdir mp3

FILELENGTH=`wc -l $1 | sed 's/[\^0-9]//g'`
COUNTER=1

while [ "$COUNTER" -le "$FILELENGTH" ];
  do
    PITCH1=`head -$COUNTER $1 | tail -1 | cut -f 4-6`
PITCH2=`head -$COUNTER $1 | tail -1 | cut -f 7-9`
PITCH3=`head -$COUNTER $1 | tail -1 | cut -f 10-12`
FILENAME=`head -$COUNTER $1 | tail -1 | sed 's/\&\&/-doubleflat/g' | sed 's/\&/-flat/g' | sed 's/\#//-sharp/g' | sed 's/\./-/g' | sed 's/\./-g'`
CLEF=`HEAD -$COUNTER $1 | tail -1 | cut -f 1`
echo $FILENAME
#make jpgs
echo "score" > mup/$FILENAME.mup
echo "rightmargin=6.25" >> mup/$FILENAME.mup
echo "clef="$CLEF" >> mup/$FILENAME.mup
echo "time=4/4n" >> mup/$FILENAME.mup
echo "" >> mup/$FILENAME.mup
echo "music" >> mup/$FILENAME.mup
echo "1: 1" $PITCH1 $PITCH2 $PITCH3 ";" >> mup/$FILENAME.mup
echo "bar" >> mup/$FILENAME.mup
mup mup/$FILENAME.mup > ps/$FILENAME.ps
gs -sDEVICE=jpeg -dJPEGQ=100 -dNOPAUSE -dBATCH -dSAFER -r300 -sOutputFile=jpg/$FILENAME.jpg ps/$FILENAME.ps
#make mp3s
echo "score" > mupformid/$FILENAME.mup
echo "clef="$CLEF" >> mupformid/$FILENAME.mup
echo "time=6/4" >> mupformid/$FILENAME.mup
echo "" >> mupformid/$FILENAME.mup
echo "music" >> mupformid/$FILENAME.mup
echo "midi all: 0 \"tempo=60\";" >> mupformid/$FILENAME.mup
echo "midi 1: 1 \"onvelocity=127\";" >> mupformid/$FILENAME.mup
mup mupformid/$FILENAME.mup
mv mupformid/$FILENAME.mid mid/$FILENAME.mid
timidity mid/$FILENAME.mid -Ow -o - | lame - -b 128 --tt triad \
mp3/$FILENAME.mp3

rm -r mup
rm -r ps
rm -r mupformid
rm -r mid

COUNTER=`expr $COUNTER + 1`
done
Appendix II: Example of a script to generate the XML code that creates drill questions in MoodleRooms

FILELENGTH=`wc -l $1 | sed 's/[^0-9]//g'`
OPTIONSFILELENGTH=`wc -l options.txt | sed 's/[^0-9]//g'`

echo "<?xml version="1.0" encoding="UTF-8"?>" > $2
echo "<quiz>" >> $2
COUNTER=1
while [ "$COUNTER" -le "$FILELENGTH" ]
  do
    QUESTIONNAME=`head -$COUNTER $1 | tail -1 | sed 's/\&\&/\-doubleflat/g'`
    ANSWER=`head -$COUNTER $1 | tail -1 | cut -f 2`
    JPGFILE=$QUESTIONNAME.jpg
    MP3FILE=$QUESTIONNAME.mp3
    echo "<question type="multichoice">" >> $2
    echo "<name>" >> $2
    echo "<text>"$QUESTIONNAME"</text>" >> $2
    echo "</name>" >> $2
    echo "<questiontext format="html">" >> $2
    echo "<text><![CDATA[<p>Identify the quality of this triad:</p>" >> $2
    echo "<p><img src="http://museprof.net/mus106/triads/jpg/"$JPGFILE" /></p>" >> $2
    echo "</questiontext>" >> $2
    echo "<generalfeedback format="html">" >> $2
    echo "<text><![CDATA[<script language="JavaScript" ...
    echo "</generalfeedback>" >> $2
    echo "<defaultgrade>1.0000000</defaultgrade>" >> $2
    echo "<penalty>0.3333333</penalty>" >> $2
    echo "<hidden>0</hidden>" >> $2
    echo "<single>true</single>" >> $2
    echo "<shuffleanswers>false</shuffleanswers>" >> $2
    echo "<answernumbering>none</answernumbering>" >> $2
    echo "<correctfeedback format="html">" >> $2
    echo "<text><![CDATA[<p>Your answer is correct.</p>]]></text>" >> $2
    echo "</correctfeedback>" >> $2
    echo "<partiallycorrectfeedback format="html">" >> $2
    echo "<text><![CDATA[<p>Your answer is partially correct.</p>]]></text>" >> $2
    echo "<partiallycorrectfeedback>" >> $2
    echo "<incorrectfeedback format="html">" >> $2
    echo "<text><![CDATA[<p>Your answer is incorrect.</p>]]></text>" >> $2
    echo "<incorrectfeedback>" >> $2
    echo "<shownumcorrect/>" >> $2
    COUNTER2=1
    while [ "$COUNTER2" -le "$OPTIONSFILELENGTH" ]
      do
        OPTIONS=`head -$COUNTER2 options.txt | tail -1`
        if [ "$OPTIONS" == "$ANSWER" ]; then
          echo "<answer fraction="100" format="html">" >> $2
          echo "<text><![CDATA[<p>$OPTIONS</p>]]></text>" >> $2
          echo "<feedback format="html">" >> $2
          echo "</feedback>" >> $2
          echo "</answer>" >> $2
        else
          echo "<answer fraction="0" format="html">" >> $2
          echo "<text><![CDATA[<p>$OPTIONS</p>]]></text>" >> $2
          echo "<feedback format="html">" >> $2
          echo "</feedback>" >> $2
          echo "</answer>" >> $2
        fi
      done
  done
COUNTER2=`expr $COUNTER2 + 1`

etc...