Digital Drum Tutor

by

Ransom Christofferson

A thesis submitted to the Faculty of the University of Colorado in partial fulfillment of the requirements for the degree of Bachelor of Science Department of Computer Science

2009

Advisor: Ken Anderson, Associate Professor, University of Colorado

Thesis Committee: Ken Anderson, Associate Professor, University of Colorado Clayton Lewis, Professor, University of Colorado Elisa Giaccardi, Associate Professor, Universidad Carlos III de Madrid

Abstract

This thesis investigates issues surrounding the design and implementation of a digital drum tutor. People have been developing digital tutors in all fields of study in an attempt to replace or complement human instructors. This drum tutor aims to replace a human instructor in some cases, but mostly to guide a student’s practice sessions, which is where the most improvement occurs. While using the drum tutor, students play predefined patterns, and in order to get the most improvement from the student, the tutor dynamically adjusts the difficulty of the pattern according to how well the student plays. Four people of varying experience tested the software. In testing, students generally found the drum tutor to be helpful in motivating them to practice more and helping them improve their playing. However, the students needed a brief explanation on how the tutor works in order to take full advantage of its capabilities.
Introduction

In the field of computer-assisted learning, many people have focused on instrumental education. As emphasized in [1], the area of instrumental learning that could benefit the most from computer assistance is daily practice. Typically, music lessons are given once a week, which gives the student the responsibility to practice during his or her own free time. This is a problem for two reasons: students often lack the motivation and focus to practice well, and if motivation is not a problem, the student may be using incorrect techniques while practicing and may not know how to practice very efficiently. If a student doesn’t practice properly, he or she can’t make nearly as much improvement as he would be able to otherwise. Computers can help give motivation by making practice more interesting, and can make practice more efficient by guiding the student’s practice session and providing feedback as to specifically how the student needs to improve.

These two topics will be the main focus of my research into designing and developing the Digital Drum Tutor. In addition, the Tutor will be able to teach a student in the complete absence of a human (for beginning drummers only).

One complication of instrumental tutoring software is that it needs to 'see' or 'hear' what the student is playing. This is, however, not insurmountable, as seen in the Violin Tutor [3], where the software uses a microphone and a video camera to observe what the student is playing. The Digital Drum Tutor will make this task much more manageable by having the student play on a MIDI-enabled electronic drum set. With this method, the computer can immediately and accurately ascertain which drum the student is hitting, and how hard.

Requirements

To briefly describe the software, the Tutor will include many lessons for a student to complete. These lessons will range in difficulty and topic, and the Tutor will allow the student to progress to more difficult lessons once the easier ones have been satisfactorily completed. Lessons will be divided into four categories: reading music (in the case of drums, this is mostly rhythm reading), technical skill (mastery over control of the sticks, i.e. being able to play quickly and accurately), musicality (ability to listen to other musicians and play along with them, changing tempo or dynamics when appropriate), and style (a student's style is revealed during improvisation and is developed by listening to and playing many different kinds of music).

During a lesson, the Tutor will analyze a student’s playing and adjust accordingly. For example, if the student is struggling to play a pattern, the Tutor will have the student try to play it slower until he/she is successful, or simplify the pattern to the point where the student is able to successfully play it. If the student is having difficulty with one specific part of a pattern, the Tutor will have the student practice that specific part. If the student is having problems staying in tempo, the Tutor will have the student work through lessons specifically designed to help with playing in time.
In an attempt to give the student the ability to practice efficiently without computer software, the Tutor will attempt to teach the student how to recognize incorrect playing on his/her own. It will do this by recording what the student plays and play it back, quizzing the student on what needs the most improvement.

The Tutor will teach on the general principle that any rhythmic pattern can be learned if you start slowly and simply. For example, to learn the basic rock pattern on a drum set, which involves both hands and the right foot, first practice the left hand’s part of the pattern by itself, starting slowly and getting faster as you become more comfortable playing it. Then repeat for the right hand, then the right foot. After that, practice playing both hands together, then the left hand with the right foot, then the right hand with the right foot. I will refer to these combinations of limbs as ‘sub-patterns’. After practicing all the sub-patterns, playing all your limbs together (the full pattern) is a much less daunting task. The Tutor will use this method very often as its main way to guide a student’s practice session. See figure 1 for a detailed description of the algorithm it uses. In the figure, one of the steps is ‘analyze the user’s playing’. This implies many calculations, which are outlined in figure 2.

To motivate the student, the Tutor will provide positive encouragement and not point out every mistake so as not to discourage the student. To provide further motivation, the Tutor will award achievements to the student (much like in the game Spore). The student will be able to see what they need to do to earn an achievement so they can anticipate and work towards a specific one if they wish, and be able to view all achievements they have earned so far. Completed lessons and achievements will earn points for the student, which they can use to purchase items such as different-sounding drum kits, which can be selected to be used during lessons (i.e., a jazz kit, a heavy rock drum kit, or a vocal percussion kit). An idea for further motivation, as discussed in [1], is the idea of a student-selected avatar or other customizations.

Some specific kinds of lessons that will be offered are: learning a new drum set pattern, embellishing and adding fills to patterns, playing snare rudiments, and playing along with a song. To describe the last lesson in that list, the Tutor will play a song (with the drum track removed), and it will be completely up to the student to come up with drum accompaniment. The student will be awarded points for playing the correct style, keeping tempo, adding embellishments and fills where appropriate, changing dynamics according to how loud the other ‘musicians’ are playing, and stopping and starting in the correct places. If the student performs poorly in one of these areas, the Tutor will point out the problem area and show the student an example of correct playing. An important thing to point out is that, in this kind of lesson, unlike the other lessons (and also unlike Rock Band), the student is not asked to play something specific. Instead, the student is allowed to exercise his/her creativity.

Most of the lessons will be presented in a linear order. However, a student may select any lesson at any time. Some features of the software will not fit into this linear lesson plan. These can be called ‘extra features’ and include the following:

- Custom patterns: The student may load in their own user patterns and the Tutor will make lessons from it, using the algorithms from figures 1 and 2.
- Metronome: Simply put, this feature just provides an adjustable metronome.
• Metronome independence: The student may choose any drum set pattern, even user patterns, and after playing the pattern for a while (without the aid of a metronome), the Tutor will let the student know how close to the tempo the student played.

• Speed playing: The student chooses any pattern, and the Tutor slowly increases the tempo of the metronome click to see how fast the student can play the pattern.

• Songs: The student may select a song from a list of many, and plays along with it. This feature of the software will not offer advice for skill improvement; it simply provides musical accompaniment.

Related Work

Similar to the Digital Drum Tutor, the Digital Violin Tutor [3] does not replace a human instructor, but is a useful tool in the absence of an instructor. The Violin Tutor uses a microphone to record a student’s playing and compares and contrasts it to a recording of the instructor’s playing, and provides very intuitive visual feedback to allow the student to more easily understand where they played incorrectly. It also teaches a student to correctly tune a violin. An additional feature is that is shows the student animations of the fingers on the fingerboard and of correct posture and physical position while playing. Instead of playing with a metronome (as in the Digital Drum Tutor), students using the Violin Tutor play on their own and the tutor transcribes the music.

The Piano Tutor [4] is also a supplement to traditional musical instruction, and like the Drum Tutor, is intended for beginners and focuses on getting the student to practice correctly. Also similar to the Drum Tutor is how the Piano Tutor uses an electronic instrument, presents multiple different lessons to the student, and may present remedial instruction in case the student plays incorrectly. The Piano Tutor also presents examples of correct playing and video lessons. However, unlike the Drum Tutor, the Piano Tutor dynamically selects lessons for the student according to how he or she is playing.

In my research, I met with Dr. Doug Walter, the professor of percussion at the University of Colorado. He pointed out that the amount of visual stimulation we receive in the world today seems quite excessive. At that point, the instruction and input the Drum Tutor used was mostly visual, so he suggested that the tutor should give some instruction or practice to learn and play by ear, for example, listening to a pattern and playing it back. A simple version of this was easy to implement in the Drum Tutor. Dr. Walter also recalled some software he had used to learn French, and how it would compliment him when he did well. The notable part was that it chose its compliments randomly from a large list. While in many cases the compliments were goofy, the randomness provided a small amount of increased interest in the software. This feature, too, was easy to implement in the Drum Tutor.
Implementation

The following features have been implemented in the Digital Drum Tutor:

- **Multiple users**: The Tutor can keep track of accomplishments and other data for an unlimited number of different users. Users can be added or deleted through the Tutor’s menu systems.

- **Pattern Lessons**: The goal of these lessons is for the student to learn a one-measure-long pattern for the drum set at a tempo which the student chooses. Each lesson starts off slowly and simply, for example, having the student start off by practicing only what his right hand would play in the pattern. The student follows along by playing with written music that scrolls by on the screen. As the student plays, the Tutor analyzes the student’s playing, showing the student how his or her playing compares to correct playing. The Tutor has the student stop playing when he or she struggles too much or demonstrates proficiency. The student obtains ‘proficiency’ when he or she plays the current pattern correctly for three measures in a row, and the student ‘struggles’ when he or she plays the pattern incorrectly for three measures in a row. As the student demonstrates correct playing, the Tutor increases tempo or has the student practice with only the left hand, then just the right foot, then combinations of different limbs, until the student is ready to play the full pattern. If at any point the student struggles too much, the Tutor requests that it decreases the tempo or simplifies the pattern by removing parts from it to a point where the student can make improvement. The student may choose to decline this offer and simply try to play the pattern again. Every pattern lesson can be played on ‘easy’ or ‘hard’ difficulty. On ‘easy’ difficulty, the Tutor is more lenient in judging the student’s accuracy in playing with the tempo and the student’s accuracy in dynamics. The Tutor keeps track of the highest tempo at which the student was able to play the pattern for both ‘easy’ and ‘hard’ difficulty. Each lesson may also be played in ‘Metronomeless’ mode, wherein the student plays the pattern for as many bars as possible along with a silenced invisible metronome before deviating from the tempo.

- **‘Play by Ear’ Lessons**: In these lessons, the student is given no visual input. The student hears a one-measure pattern and is asked to play it back. These lessons are designed to train a student’s musical ear and aural recognition of rhythms and the different parts of a drum set. Unlike the pattern lessons, after the Tutor gives one bar of clicks at the start, the student only has one measure to play back the pattern. After this one measure, the Tutor assesses the student’s playing. If the student plays incorrectly, the Tutor shows which notes were correct and which were incorrect, but does not visually show the correct pattern or simplify the pattern to make it more easily playable.

- **Custom Lessons**: A student may construct any single-measure drum pattern, and the Tutor creates a ‘pattern lesson’ from it. The student may choose ‘easy’ or ‘hard’ difficulty for a custom lesson, or play it in ‘metronomeless’ mode.
• Rudiment Lessons: A rudiment is a basic pattern used in drumming, usually played only on one drum. The Tutor has the student use two different drums, one for the left hand and one for the right hand so it can differentiate between the two hands. The Tutor has the student start out by playing the rudiment slowly and increases tempo as the student improves. The rudiment lessons work very similarly to the pattern lessons.

• Randomized Encouragement: Every once in a while, when the student performs well, the Tutor gives positive encouragement. The compliment that the Tutor gives is randomly selected from a list of many in order to keep the experience of using the Tutor less monotonous.

• Achievements: Achievements provide specific goals for the student to work towards, for example, “play at least 5 bars in Metronomeless mode” or “complete all lessons on easy difficulty.” The student is allowed to see the requirements for every achievement, and when a student completes one of those goals, he is congratulated and this accomplishment is represented by allowing the student to see the title of the achievement and a shiny graphic that accompanies it on the ‘achievements’ screen.

• Shop: As lessons are completed, the student earns credits which he can use to purchase items in the shop, such as different sets of sounds that the drums make when they are played.

Some of the above features were not in the original plans for the Drum Tutor, such as the ‘Play by Ear’ lessons and the randomized encouragement, and many features originally planned were not implemented, due to time constraints. The Drum Tutor does not currently teach how to read standard musical notation, does not provide lessons for the student to develop his/her own style (style is reflected in improvisation), and it only touches on developing a student’s musicality (learning to play with other musicians and being attentive to dynamics and how the music sounds overall) through the ‘Play by Ear’ lessons. Another feature originally planned but not implemented was the ability for the student to play along with songs.

Figure 1 shows the proposed algorithm for a pattern lesson. This was implemented almost exactly as proposed; there are only a few differences. Before a lesson begins, the student sets a starting tempo and ending tempo. During the lesson, the student works his/her way up to the ending tempo, and when the student plays the full pattern at that tempo, the lesson ends. The proposed idea was to have the tempo gradually speed up at that point until the student makes a mistake. The second difference is that the Drum Tutor was not implemented with the ability to detect correct playing at the wrong tempo. The last difference is that no ‘special lessons’ were implemented, which would be special instruction unique to each pattern lesson given to the student when the student struggles in such a way that the Tutor cannot provide any other suggestions to fix the student’s erroneous playing. The correction the Tutor provides instead of a ‘special lesson’ is just to slow down the tempo.

Figure 2 shows how the Tutor determines in what manner the student is having difficulty. This algorithm is implemented as-is, with the exception that the Tutor cannot determine whether a student is playing the pattern correctly but not in the right tempo.
The Digital Drum Tutor was created primarily with Adobe Flash. Flash was chosen because of my former experience and skills with it and because it provides an easy way to incorporate graphics, animation, sounds, and interactivity with robust code. The downside to using Flash is that it was developed primarily for the web, and as such, has no inherent ability to communicate with MIDI devices or write to disk. These two problems were solved by introducing C++. The Drum Tutor is a C++ application with an embedded Flash object. The C++ code receives the signals from the electronic drum kit and passes the data to the Flash object. Since Flash cannot write to disk, when the Tutor saves information, the Flash object passes data to the C++ code, which then writes it to an SQLite database.

Although Flash has simple built-in functions to control sound, this sound does not behave consistently across all machines. On most Windows computers, there was a considerable (at least half a second) delay between when code to play a sound was executed and when the sound actually came out of the speakers. This is a rather large problem for software that relies on playing an instrument with accurate timing. C++ also has built-in controls to play sounds. Unfortunately, this method did not prove to be a more viable solution. I explored a couple C++ audio libraries; the first one I tried is called ‘BASS’ [5]. With this sound library, there was only a small (but still noticeable) delay in sound even when playing the sound from memory instead of from the file on the disk. The second sound library I tried, FMOD, did not prove to be successful, as there was an even larger delay in sound. Judging by the reviews I read, FMOD probably would be able to perform to my satisfaction, but I could not get it to do so after the many hours I spent with it.

Another small difficulty I encountered was physically hooking up a computer to the electronic drum kit to receive its MIDI signals. Most computers do not have MIDI ports. Many older computers have joystick ports (also called game ports) which easily double as connectors for MIDI instruments. Most computers do not have these nowadays, but cheap sound cards with joystick ports on them can be purchased and installed. I tested this method out and it works, but it is not a very practical solution for the average person, as it requires the installation of a soundcard. An easier solution to this problem is to use a USB-to-MIDI converter. This is a small USB device that allows a computer to send and receive MIDI signals through a USB port. This is an easy solution; however, the small delay in sound may be due to the processing that occurs within the USB device.

The electronic drum kit that was used in developing the Tutor has many features – for example, the edge of the ride cymbal, when hit, will send a different MIDI signal than the bow of the ride cymbal. This is to enable the different sounds that this would make on a real drum kit. However, this is not always important to the Drum Tutor. A beginning drummer should not focus on what part of a cymbal he or she is hitting, but focus instead on playing the correct rhythms. The Drum Tutor takes advantage of the drum kit’s many features and at the same time focuses on what is important to the student by using inclusion polymorphism. For example, the Tutor may want the student to play a note on the ride cymbal, but not care whether the student plays on the bow or the edge of the cymbal. If the student plays on the bow, an object of type ‘RideBowHit’ is created. If the student plays on the edge, an object of type ‘RideEdgeHit’ is created. The Tutor checks to see if what the student played is a correct hit by evaluating whether or not the object created is a ‘RideHit’, which is a superclass to both ‘RideBowHit’ and ‘RideEdgeHit’. This class structure allows for either the bow or edge to be correct. In a different circumstance, the Tutor might require the student to specifically play on the edge of the ride cymbal. This requirement is easily implemented in light of
the class structure used for all the different notes that can be played on the drum kit, which is diagrammed in Figure 3.

A ‘hit’ object is created each time the student hits part of the drum kit, and when the Tutor evaluates the student’s playing, it compares those ‘hit’ objects to a predefined set of ‘hit’ objects which compose a drum pattern such as a rock beat. Each ‘hit’ object has three attributes: velocity, timestamp, and limb. If the ‘velocity’ and ‘timestamp’ values are very similar for the hit that is played and the hit that is expected, that hit is considered to be correct. The ‘limb’ attribute is used to keep track of which limb (i.e. right hand, left foot) should be used to play a certain hit. The value of this attribute determines how the Tutor simplifies a drum pattern when the student struggles. In some cases, the Tutor will simplify the pattern by removing what the right hand would play, or only requiring the student to play the left hand’s pattern.

In total, the Drum Tutor represents more than 3900 lines of code written in Flash (as well as graphic design and animation work not represented by lines of code), over 600 lines of code in C++, over 60 classes in Flash (31 of which are ‘hit’ classes as described earlier) and several more classes in C++.

**Evaluation and Results**

A total of 4 students tested the system: two absolute beginners who are good friends of mine, one moderately new drummer who I had been giving lessons to for a couple months, and one drummer with over 7 years of experience. Within one hour, one of the absolute beginners was able to play a basic rock pattern at 80 beats per minute and completed the first ‘Play by Ear’ lesson. The other beginner became frustrated with her inability to play during use of the system, but nonetheless managed to successfully complete the basic rock pattern in a similar amount of time as the first student. This completion of lessons shows a tangible result of progress and was encouraging for the students. The moderately new drummer found the Drum Tutor to be very useful in learning new patterns. In the past, she often found herself wanting to practice drums, but she was unsure what to play, and the Tutor very explicitly provides a solution to this. The experienced player found the system to be fun, but as more of a novelty, with the exception of the ‘create a lesson’ feature, since the lessons are geared towards newer players.

The beginner students expressed a wish that they could play better so they could earn more achievements and credits, which shows that the Tutor provides motivation to practice. These students also found that they sometimes had more success in playing through the lessons when they did not look at the screen when they played, and instead just listened to the metronome and focused on what they were playing.

Some brief explanation had to be given to the students while they used the Tutor, since the Tutor does not immediately explain some things on its own, such as how to earn credits or roughly how the Tutor organizes and progresses through lessons, and how they can control the flow of a lesson.
Limitations

There are some shortcomings to the software, partly due to time constraints in development. The Tutor does not completely teach how to read music. It demonstrates which symbols on a musical staff correspond with which drums, but it does not teach rhythm reading. This makes it necessary for an instructor or other source to teach this to the student. In many cases, the Tutor is able to point out how the student is playing incorrectly, but it is not able to provide a specific exercise to fix the student’s incorrect playing. Instead, it slows down the tempo or simplifies the pattern. Furthermore, in every lesson, the student must play in time with a metronome. The timing of the student’s playing is directly compared to the timing of the metronome, and the difference in timing determines whether a note was correctly or incorrectly played. This creates a severe limitation in the software if the student is unable to play along with a metronome. On the other hand, for a drummer, being able to play in time is extremely important and needs to be learned well, and this requirement to always play in time with a metronome puts a strong emphasis on learning correct timing.

These shortcomings make the Tutor unsuitable for beginners to use without the addition of a musical instructor and it does not properly prepare students to play with other musicians. However, it is still a very handy tool for students who have received basic instruction for the drum kit.

Future Work

The Digital Drum Tutor could greatly benefit from the addition of all the features that were originally planned in the design but never implemented. Another nice addition would be to integrate video lessons before or at times during a lesson. Feedback from prolonged use of the Tutor from beginning students would be quite helpful in determining exactly how useful the current features are.

The required equipment for the Drum Tutor is expensive; a very low-end MIDI-enabled electronic drum kit often costs at least $400. Fortunately, the popularity of the ‘Rock Band’ and ‘Guitar Hero’ games has put cheap electronic drum kits in the homes of many people. These drum kits don’t have as many triggers to hit and can’t determine hit velocity, but the Drum Tutor could be adapted to work with these common kits.

Conclusions

Overall, this was a fun project. I learned that many programming languages can be combined into one system, and that a large programming project will take longer than you expect. I also learned a lot about Adobe Flash and that computers are not yet ready to replace human tutors.

My research in this field demonstrates that a digital instrumental tutor does not need to be very complex in order to be very useful. Compared to the Piano Tutor and the Violin Tutor, the Drum Tutor has a simpler method of assessing a student’s playing and uses the same algorithm to guide most of its lessons.
From my studies, I have concluded that digital instrumental tutors will not replace human instructors until someone creates a tutor that can hear the student play and be intelligent enough to determine the skill level of the playing, see if they are using correct technique, administer appropriate exercises, and be able to identify any problems a student could encounter and provide an appropriate solution.

References


Figure 1
Figure 2
Figure 3
### Screenshots

**The ‘Select Lesson’ screen**

![Select Lesson Screen](image)

#### Main Lessons

<table>
<thead>
<tr>
<th>Lesson</th>
<th>High Scores</th>
<th>Easy</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Rock</td>
<td>120</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>More Bass Drum</td>
<td>100</td>
<td>170</td>
<td></td>
</tr>
<tr>
<td>Crash</td>
<td>120</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>Paradiddles</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Funk</td>
<td>130</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Swing</td>
<td>120</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- Create a lesson
- Extra Lessons
- Back
A lesson in progress
The ‘Achievements’ screen