SOFTWARE FOR MUSIC EDUCATION

OFFKEY STUDIO:

Software for Music Education

By

Duy Quoc Nguyen

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelor's degree

With Honors in

Computer Science

THE UNIVERSITY OF ARIZONA

MAY 2008

Approved by:

Suzanne Westbrook
Department of Computer Science

STATEMENT BY AUTHOR

I hereby grant to the University of Arizona Library the nonexclusive worldwide right to reproduce and distribute my thesis and abstract (herein, the “licensed materials”), in whole or in part, in any and all media of distribution and in any format in existence now or developed in the future. I represent and warrant to the University of Arizona that the licensed materials are my original work, that I am the sole owner of all rights in and to the licensed materials, and that none of the licensed materials infringe or violate the rights of others. I further represent that I have obtained all necessary rights to permit the University of Arizona Library to reproduce and distribute any nonpublic third party software necessary to access, display, run, or print my thesis. I acknowledge that University of Arizona Library may elect not to distribute my thesis in digital format if, in its reasonable judgment, it believes all such rights have not been secured.

SIGNED:

ABSTRACT

Learning to play musical instruments can require a large investment of both time and money for individuals since instruments can be expensive to buy and tend to have relatively steep learning curves. This paper describes one student's senior honors project: a software application called OffKey which aims to improve accessibility and education in music by allowing individuals to use a computer keyboard as an instrument for playing and composing music. The OffKey project encompasses both OffKey Studio, which facilitates music composition, and OffKey Play, which teaches reading music.

Introduction:

As a Computer Science student, my interests have led me to independently develop software for my Honor's Thesis Project. The purpose of the project is to create software outside of the classroom environment in hopes of applying knowledge acquired at the University, while at the same time introducing myself to topics that are not presented in class. For this project, I decided to cross subjects, interweaving my interests in music as well as computer science. This allowed me to produce software for a non-typical audience. While courses at the University focus on producing software for professionals in the field, this project requires producing software for a wider demographic.

The project is coded in Java. I chose to use Java for various reasons. Java's portability makes it easy to develop software for multiple platforms, allowing the software to be accessible to virtually anyone. At the same time, the vast amount of documentation and software libraries speeds up development. I am also most comfortable using Java, as it is commonly used in various assignments in my computer science courses. Through this project, I refined my programming skills, as well as developed new skills in user interface design, animation, research, and time management. Not only did I complete two applications during the course of the thesis, but I also submitted a paper to the 2008 Conference in Innovation and Technology in Computer Science Education (ITiCSE).

Premise:

I believe that music is important, and I am sure that many people agree with me. Not only is music enjoyable, but listening and experiencing music (any kind of music) has been attributed to improve cognition and development [7]. Despite that, relatively few people have actually taken the time to learn to play a musical instrument. Lack of time and money can account for some loss of motivation. While many people would love to learn to play their favorite songs instead of simply listening to them, the climb to becoming a musician can be steeper than most people are willing to endure.

This project, called OffKey, is an attempt to make that climb easier. The idea is to take a concept that people are already familiar with and use it as a bridge to accelerate the learning process. The object of interest is the computer keyboard. Computers are common in today’s world, and many people have learned to type as necessary for their work, study, or recreation. The idea behind the OffKey Project is simple: turn the computer keyboard into a musical instrument through software.

The OffKey Project comes in two parts. The first part is the actual instrument itself, called OffKey Studio. When the application is running, each key on the computer keyboard is assigned a musical note. When the user presses a key, the corresponding note is played through the computer speakers. Through this software, users can change how keys are mapped, select which combination of sounds are played, and watch how their notes appear on a musical staff. With only a simple download, users have a fully functional musical instrument at their fingertips with no additional costs past the personal computer sitting in front of them.

However, learning to play this musical instrument can still be difficult. As such, OffKey Studio is accompanied by its sister application OffKey Play. OffKey Play is meant to be a personal musical trainer, developed in the form of a video game. Taking the “rhythm game” format of commercial works like Dance Dance Revolution [3] and Guitar Hero [4], this game prompts the player to press certain keys and scores the player based on accuracy. The player selects a song froma list, then selects an instrument to play. The game then displays a series of scrolling musical notes on a musical staff, each marked with a character found on the keyboard that corresponds to that note. When the note reaches the left side of the screen, the player presses that key, and the note is played through the speakers. Any accompaniment is played automatically while the user plays. The player has an unlimited number of songs to choose from, as the software uses MIDI files to generate the notes. By using OffKey Play to practice, users develop a sense of rhythm and muscle memory that will help them play more difficult songs. Soon enough, players will become accustomed to using their computer keyboard as a musical instrument, reaching the transition to becoming a musician.

Beginnings:

The OffKey Project started as a fuzzy idea around my sophomore year at the University. Rhythm games have been around for quite a while by this point, simulating everything from dancing to guitar. Having played these sorts of games since the early days of imported Dance Dance Revolution machines, I am very familiar with their basic mechanics and their effects on people. I have noticed that those who play these games acquire a greater appreciation for music, a stronger sense of rhythm, and the ability to “decompose” music (for example, someone who plays a simulated bass guitar regularly tends to pick out the bass out of a song easier). Already, people are experiencing the intricacies of music without having to touch much of the textbook theory behind it. OffKey is an attempt to go one step further.

The ideas around the OffKey Project did not really take form until I received my imported copy of Daigasso! Band Brothers [2], a rhythm game for the Nintendo DS that was never localized for the United States. I feel that this game is very different than most other music games I have played. While in most other rhythm games the button combinations are purely arbitrary and have very little relation to the musical note the player hears, Band Brothers assigns each of the eight face buttons on the system a specific note that would be played whenever the player presses that button. The player can adjust the pitch of the note by half a step by using one of the shoulder buttons, or raise the note a full octave higher with the other. The result is a handheld musical instrument that was easy to play because of my established familiarity of the Nintendo DS control style.

I started planning OffKey based on the same concept, except using the familiar QWERTY keyboard as the musical device. At the time, I had finished the instructional courses that would allow me to start programming my project. I managed to finish a Java-applet as a proof of concept during the summer after my sophomore year. The applet would play piano sounds as the user typed keys. Much the coding used Java's fairly primitive Midi libraries, which made it very difficult to change instrument sounds. When summer ended, the project was put on hold due to time restraints.

The project was picked up again near the end of my third year. I was adamant in turning the OffKey Project into my honor's thesis. It would be a great way for me to allocate time in completing the project, while receiving feedback and criticism from a faculty advisor. With the help of my thesis advisor, Suzanne Westbrook, I put together a rough agenda of how the project would progress the following year. The Fall semester of 2007 would be dedicated to the development of OffKey Studio. The following Spring semester would be dedicated to the development of OffKey Play, as well as the final thesis paper. The entire project would be complete by May 2008.

OffKey Studio:

Over the summer of 2007, I worked on a second applet, this time using the more developed Java Sound library [6]. This gave me a chance to experiment with Java's new updates, as well as make considerations to the design of the final product. Java Sound sped up the development quite a bit by offering both a higher quality sound collection and a more efficient programming environment. I was also working on putting together my own website, DashWorks Studio [9], as a place to showcase my software (among other things). This would also be the home for the OffKey Project, where I can host the applications as well as gain feedback from future audience.

The first semester of my senior year was dedicated to coding and documenting OffKey Studio. Much of the time was spent on developing a few libraries that would make some basic music functionalities (such as playing notes and changing volume) simpler. I managed to add a few interesting features, such as the ability to play multiple instrumental sounds at once, and displaying played notes on a Grand Staff.   I finished the first version of OffKey Studio late October, and continued to polish it up until late December.

I was surprised how straightforward it was to put together this application. The only real difficulty I came across was learning to program with Java Sound. I found that I have developed certain programming habits through my years at the University. Most of the programming knowledge I used came from my classes in Programming Design/Development (C SC 227) and Object Oriented Design (C SC 335) during my first year at the University. Concepts such as event driven programming were essential when developing software with a graphical user interface. One thing that really surprised me was my automatic implementation of common Java Programming Patterns, a concept that I thought unimportant while in class. Almost without realizing, I implemented the Model-View-Controller design into OffKey Studio. Later on, I found myself implementing the Strategy design into OffKey Play to handle when notes should be played. By the end of it all, I found that all knowledge, regardless of how useless it seems at the time, might become useful later.

While Java Sound was helpful in completing OffKey Studio in a reasonable time, it was still quite difficult to work with. While Java provides a large amount of information in its JavaDoc system, the Java API [5] only really tells programmers what each of the functions do. I had a lot of trouble figuring out how to fit the different classes together to get some very basic functions. I was lucky enough to find the Java Sound Resources [10], which provided sample code that allowed me to piece the seven different Java classes together to properly play a single note. Friends who have helped demo the software have reported problems with Java Sound on systems with multiple sound cards and other non-typical conditions. Unfortunately, I only have a Windows based machine to perform my tests. Tackling problems with these other systems has been a slow process, but necessary to achieve the level of accessibility I desire. There did not seem to be an alternative to Java Sound, apart from redeveloping the library from scratch, so I continued to use Java Sound to achieve results on the majority of computer systems available.

A Java software editor called NetBeans IDE [8] was a huge time saver when it came to designing a user interface. Previous experiences with coding boxes and buttons in Java manually have been a bit more than discouraging. Java Swing (Java’s built-in interface library) is notorious for being difficult to use. It is sometimes hard to visualize the placement of textboxes and buttons when working in purely code. At the same time, Java has a habit of making its own decisions when it comes to the location of these items. Many of the parameters the programmer sets (such as the size of the window) are merely suggestions, and Java has final say as to where things are placed, leaving the programmer with a less-than-adequate layout. This is problematic when I am trying to achieve an intuitive interface for my software. NetBeans, however, allowed me to drag and drop my items, saving me a lot of time and frustration at the cost of ugly automatically-generated code. There is no doubt that I will continue to use NetBeans for most, if not all, future software.

ITiCSE:

During the semester, Suzanne mentioned an annual conference on Innovation and Technology in Computer Science Education. While my software was intended for education in music, not computer science, she urged me to consider writing a paper for the conference under the "Undergraduate Capstone Projects" category. During the winter break and the early part of the Spring 2008 semester, Suzanne and I put together a full research paper describing the software and its intentions.

Writing the paper forced me to look outward and consider what others have done in the field related to the OffKey Project. Through research, I have found studies done on the affects of video games on retention and how music can be beneficial to learning and development [7, 12, 13]. I have also found projects that have similar philosophies, such as Piano Wizard [11], which attempts to teach children how to play the piano using a Midi Keyboard connected to a PC. Though the research, I learned how to compose a proper paper and what steps I needed to take to submit it. I also learned that OffKey is not an isolated project, and that I should take advice from other researchers to improve my software.

I would have to say that I never enjoyed writing a paper as much as I did this one. After a few very late nights, and lots of caffeine, I was able to finish the paper and submitted it for review. Unfortunately, the paper was not accepted to the conference, primarily due to the scope of the project (the software had little to do with Computer Science education, a brick wall that we had already anticipated). Regardless, I still have a completed research paper that is ready for editing and resubmission to a different conference in either Computer Science or Music

Education. The paper became a meaningful addition to the overall project that can still be useful to me sometime in the future.

OffKey Play:

OffKey Play was the last part of the project to complete during the Spring semester. This part of the project turned out to be a real test of my skills as a programmer. While OffKey Studio involved using many of the more familiar concepts of Java programming, OffKey Play required better knowledge of some of the more difficult concepts of programming. Not only did I need to learn how to parse Midi files (Java Sound Resources provided a lot of help), but I also had to learn how to handle threading, graphics, and animation.

Threading and concurrency was a huge factor in completing OffKey Play. The game was very time sensitive, since players must press the right keys at the right time. This required handling multiple threads to monitor different parts of the program at the same time. This includes updating animation frames, scoring, and playing background music. Unfortunately, I enrolled in a Parallel Programming course the same semester, so my knowledge about threading was relatively thin. With some pointers from a few friends and strangers over the internet, I managed to synchronize these jobs properly.

Graphics and animation were also challenges, as I have only used Java's Graphics2D library briefly in my freshman year. While I played with 2D animation back in my sophomore year of high school, using Java was much different. I ran into many difficulties, such as graphics skipping, objects constantly blinking, major slowdowns, and graphics going off sync from the music. Unfortunately, many of the tutorials on the internet are very outdated, so much of my time was spent on trial and error. With some help from About.com [1], I managed to achieve relatively smooth animation at 20 frames per second using some buffering techniques. While 20 frames per second is perfectly fine for the functionality of the software, it is still relatively poor considering I am only using basic 2 dimensional graphics.

It was not until recently that I found Java3D as an alternative graphical solution. It seems that Java2D is simply obsolete, and the Java3D library can produce smooth 2D and 3D animations more efficiently. While I did not have time to implement Java3D into the current release of the software, plans on improving the animation quality of OffKey Play are well underway.

Conclusion:

Both applications are available at my personal website, Dash Works Studio [9]. The spoils of the OffKey Project are as follows: 2 applications with a total of 4,849 lines of Java across 23 classes. OffKey Studio is available in a 448kb compressed rar file, and OffKey play is available in a 408kb compressed rar file. Other products of the project include one website written in HTML and Python, one 5-page research paper, and one Honor’s Thesis paper.

A project such as OffKey, can never really achieve a point of success. There is always work to be done, and I intend to continue improving the software. This thesis only marks a checkpoint in the development of the software. There is plenty of work that needs to be done until I am satisfied with the state of the OffKey Project. However, for that I need more practice in coding. I need to do research in regards to my coding methods and the technologies available to me. I need feedback from a set of future consumers, and I need more time than what can fit within two semesters.

Works Cited

[1] About.com, 2008. Retrieved February 20, 2008 from The New York Times Company: http://www.about.com

[2] Daigasso! Band Brothers. Kyoto, Japan: Nintendo of Japan, 2006

[3] Dance Dance Revolution. Osaka, JP: Konami (1998).

[4] Guitar Hero. SunnyVale, CA: RedOctane (2005).

[5] Java Platform SE 6 API, 2006. Retrieved April 22, 2008 from Sun Microsystems: http://java.sun.com/javase/6/docs/api/

[6] Java Sound API, 2008. Retrieved January 6, 2008 from Sun Microsystems: http://java.sun.com/products/java-media/sound

[7] Music Thought to Enhance Intelligence, Mental Health, And Immune System, June 22, 2006. Retrieved January 2, 2008, from ScienceDaily: http://www.sciencedaily.com/releases/2006/06/060622172738.htm

[8] NetBeans IDE. Retrieved April 22, 2008. http://www.netbeans.org/

[9] Nguyen Duy. DashWork Studio, November 21, 2007. Retrieved January 6, 2008 from http://www.dashworkstudio.com

[10] Pfisterer, M. and Bomers, F. Java Sound Resources, February 17, 2005. Retrieved January 6, 2008 from http://jsresources.org

[11] Piano Wizard, (N.D.). Retrieved January 6, 2008, from Allegro Multimedia Inc.: http://www.pianowizard.com

[12] Rebetez, C. and Bentrancourt, M. Video Game Research in Cognitive and Educational Sciences. Cognitie, Creier, Comportament/Cognition, Brain, Behavior. 11 (1). 131-142

[13] Stewart, B. Ezell, S., DeMartino, D., Fifai, R., and Gatterson, B. Virtual Technology and Education. Quarterly Review of Distance Education. 7 (4). 377-385

OffKey: Introduction to Music through Computer Science

Duy Nguyen
University of Arizona
2735 North 15th Avenue
Tucson, Arizona 85705
1-(520)-204-5382
dqn@email.arizona.edu

Suzanne Westbrook
University of Arizona
Department of Computer Science
Tucson, AZ 85721
1-(520)-626-9196
sw@cs.arizona.edu

ABSTRACT

Learning to play musical instruments can require a large investment of both time and money for individuals since instruments can be expensive to buy and tend to have relatively steep learning curves. This paper describes one student's senior honors project: a software application called OffKey which aims to improve accessibility and education in music by allowing individuals to use a computer keyboard as an instrument for playing and composing music. The OffKey project encompasses both OffKey Studio, which facilitates music composition, and OffKey Play, which teaches reading music.

Categories and Subject Descriptors

J.5 [**Arts and Humanities**]**:** Performing Arts; K.3.1 [**Computers and Education**]: Computer Uses in Education – *computer assisted instruction (CAI)*

General Terms

Design, Human Factors

Keywords

OffKey Studio, OffKey Play, Music, Education, Java, Games

1. INTRODUCTION

Although music has been attributed to improving cognition, intelligence, and overall health [5], introducing oneself into the field of music can be very difficult. There is a large investment of both time and money for individuals who are interested in learning to play a musical instrument. Musical instruments tend to have relatively steep learning curves. At the same time, the initial costs in purchasing a musical instrument (and possibly the cost of hiring an instructor) act as a large deterrent for those who wish to learn to play music casually. Foregoing classes, self study can be arduous and frustrating.

This paper describes one student's senior honors project: a software application called OffKey. The project is the student's own conception and design with the goal of allowing Offkey users the opportunity to become better acquainted with music, both in composing and in learning to read musical scores, without financial and accessibility barriers. While the software is meant for recreation, it has an educational side effect that the user may or may not initially realize. For the honors student, the project pulls together skills acquired during his undergraduate studies as well as an opportunity to learn independently about areas not studied during coursework such as human-computer interaction.

2. GOALS

2.1 Software Goals

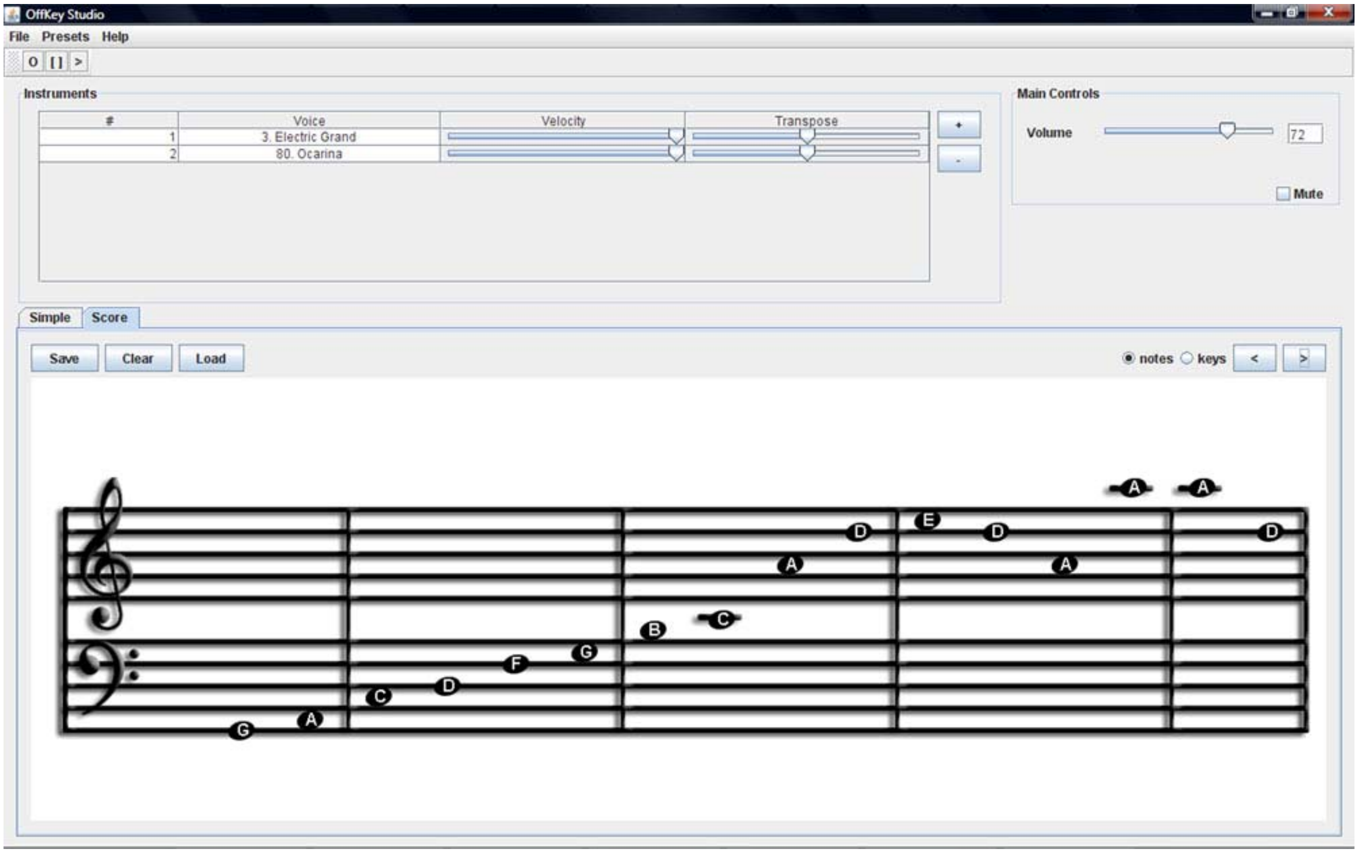
The goal of the OffKey project is to reduce the barriers that deter people from learning music as much as possible. We believe music should be made more accessible, allowing a large and varied demographic to learn to play and compose. The idea behind the project is to create a new instrument that can be used as a stepping stone into the realm of music. This new instrument should be affordable, easy to obtain, and easy to learn. As such, it should use equipment and utilize skills many people have already acquired.

In this age, computers are almost a necessity for work, education, and entertainment. Most people have access to computers with basic internet service, whether in their own homes or at a public resource, such as a library. These computers can be used as platforms to create a new musical instrument. By providing a free application that serves as a software-based instrument, users can learn to play music using computer hardware they already own. The application will be offered free of charge, completely removing any additional spending other than obtaining initial access to a computer.

Along with the software that will provide the instrument, there should also be additional software to help people learn how to use the instrument in a manner that is both intuitive and effective. One idea is to develop software in the form of a video game. Although research in the field of video games is still relatively new, many argue that interactive simulations can have many different effects on the players, some of which can be used as a means to teach [9]. Others argue that an association between video games and course work can improve performance because players can connect and interact in ways in which they are accustomed [11]. This game will also be offered free of charge, keeping consistent with the project’s goal to reduce costs.

2.2 Student Goals

The OffKey Project is an experiment in interacting with users who are not necessarily programmers. When studying at a University, most of the software a student writes is evaluated by a professor or an assistant who is a professional within the field.



#	Voice	Velocity	Transpose
1	3. Electric Grand		
2	80. Ocarina		

Volume 72

☐ Mute

● notes ○ keys

Figure 1. The OffKey Studio Interface

Thus, the software is written in a form and works in a way that is understandable to these professionals. However, consumers may not have expertise in the field of computer science, and may not be able to interact with the same software just as easily. This project is an attempt to communicate with users to create interactive software that is intuitive to use and is approachable by people of different backgrounds.

The project is also a chance to express creativity in software development. In the classroom, projects tend to have very strict specifications. The project's functionality is well defined, as well as the testing and grading, because professors have refined their assignments over their years of teaching. Independent development is very different because the end users need to be satisfied, but the developer is unaware of all of the users' expectations while designing the software. This project is not only an exercise in programming and design, but also time management and customer support.

Apart from the actual software itself, documentation is needed. The project includes designing and maintaining a website that can be used to distribute software, communicate with the user base, and offer tutorials, support, or other insights that may be interesting to the consumers.

3. IMPLEMENTATION

OffKey Studio and OffKey Play are attempts at developing software that fulfills the ideals of the OffKey project. The software is written in Java, utilizing the Java Sound library [4,7] to produce MIDI-like functionality. Using Java allows functionality across many computer platforms, further improving accessibility.

3.1 The Instrument: OffKey Studio

The physical instrument is the core of the entire project. Simply stated, OffKey Studio will use a computer to turn any QWERTY keyboard into a musical instrument. A note is mapped to each key so that when a user presses a key, that particular note will play through the computer's speakers. The software uses a skill that many people have (i.e., typing) by translating similar finger movements to produce music.

OffKey Studio includes many of the functions expected of a MIDI piano, including a wide selection of voices (based on General MIDI), sustaining, and pitch bending. Multiple voices can also be set to play in unison, allowing users to blend multiple instrumental sounds together. Users can also configure presets to save and load their favorite combinations of sounds. This allows users to switch between instruments seamlessly while they perform.

Visualization of the music is also provided. A grand staff is presented at the bottom half of the screen, and musical notes appear on the staff as the user presses keys. The user can scroll to see the notes that have been logged, and can even save the log of notes to view later. The user also has the option of displaying characters on the note (either the keyboard characters or the musical letter notation). This aids in associating key presses with musical symbols.

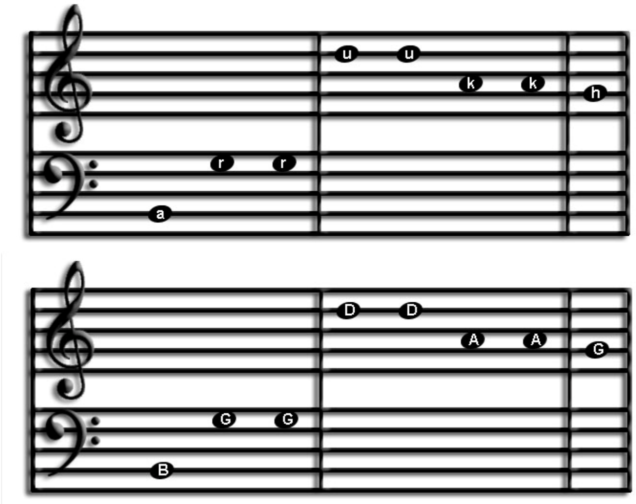


Figure 2. Users can specify if they want to see keyboard characters (top) or musical letter notation (bottom)

An important consideration in developing the software is the layout. While various software emulate a piano[1] [8,10] (where the white keys are placed on one row and the black keys are set on the row above), OffKey Studio allows users to define their own layouts. With an arbitrary layout, the keyboard can sport three to four octaves. Experiments with different configurations are being done to see which layouts are easier to use. It is possible that different users would prefer their own layouts, similar to how many guitarists have custom tunings to their instrument.

OffKey Studio will allow many people to instantly have their own musical instrument at their disposal by simply downloading the software from its website [6].

3.2 Learning: OffKey Play

Music based games (e.g., Guitar Hero [2], Dance Dance Revolution [1]) have already achieved much popularity amongst a wide demographic. These games reward players with points when they perform a sequence of actions (such as strumming the fake guitar or stepping in the proper position) in a particular rhythm, based on the songs they choose to perform. As players become skilled at the game, they can attempt to perform more complicated sequences.

The principle of the OffKey learning software is the same. Using similar mechanics, OffKey Play can help introduce users to playing the instrument, and help them become skilled at playing more complicated musical pieces as they improve. At the same time, the player becomes accustomed to reading sheet music through the game's interface.

The premise of the game is simple. A grand staff is fixed on the screen and a set of musical notes corresponding to a chosen song will scroll towards the left. Each note is marked with a character that corresponds to the correct note. The player must press the right key(s) on the keyboard when the note reaches the left end of the staff. The player is then scored by how accurately the song was played. Difficulty varies depending on the song the player decides to perform. The application uses the MIDI interface to generate notes that appear on the screen. This allows users to download their favorite songs in MIDI format from various sources online, and perform those songs in OffKey Play.

The application should be enjoyable while users become accustomed to using their keyboard as a musical instrument. The game format should aid in retention, and the point system should motivate users to improve over time. Also, the player learns to read sheet music through the software due to using musical notation as prompts. A toggle allows the users to disable the letters marked on the notes. The result is the ability to play music using printed sheet music. This skill can be invaluable, as the ability to read sheet music can ease the transition from OffKey to other instruments.

The final result should be an educational and recreational experience. The game can be played in short doses or in wide spans, depending on the user's own time. The software makes learning to play a new (or, possibly, first) instrument less frustrating. By experiencing music first hand, users may become less likely to defer learning and become more motivated to play, while picking up on theory and notation while they play, instead of beforehand.

4. SIMILAR WORKS: PIANO WIZARD

Piano Wizard [3,8] (and its sister application, Guitar Wizard) aims to teach people how to play music in a manner very similar to OffKey Studio. After installing the software, the user connecs a MIDI keyboard (or custom guitar peripheral) to their computer and plays a video game that involves colored eggs scrolling upwards towards a color-coded piano on the screen. The player presses the corresponding key on the physical piano when the egg reaches the top. The software comes with stickers that color code the physical piano to help users find the right key quicker.

The OffKey Project has the same intentions as Piano Wizard; to allow users to experience music before learning the theory, in hopes of inspiring users to pursue further education. However, Piano Wizard fails to eliminate the financial barrier with a bundle pack (which includes software and a MIDI piano) for $200, or a stripped version of the software on its own for $50 (in which case, the user would have to purchase a piano separately).

The software is also marketed towards children with its art direction and elementary school song selection, although adults can learn as well by loading their own MIDI files. Aside from the initial monetary investment, Piano Wizard may be enough for many people to learn to play a piano without too much frustration.

5. FUTURE WORK

The OffKey project is still a work in progress. While early versions of the software are available online, both the instrument and learning applications are still being polished and streamlined based on the needs of users. OffKey Studio will have better visualization, which will include logging chords and note duration. User will also be able to change time and key signatures, set tempos, and playback their logs in different instruments.

[1] Piano Wizard has an option to use a typewriter keyboard in place of a MIDI piano.

Supplemental applications for the project include a key layout editor, allowing users to visualize their key layouts and making it easy to modify the layout to their tastes.

Plans also include fixing bugs and improving the user interface. While documentation on using the software is available, the goal is to make the software intuitive enough where using the documentation is minimal.

Once the key parts of the software are finished, work will be done on other applications related to the OffKey project. Current ideas include developing software that will help users of OffKey Studio compose their own music. There are also plans for extending the instructional game to include network multiplayer support. This would allow multiple people to connect computers on a LAN or through the internet to play songs in unison. It would be possible for many people from around the world to connect and play a digital symphony.

6. CONCLUSION

The goal of increasing accessibility to music through software has not been met. The applications are still very young, and the user base is still very small. Currently, news about the software has only been passed through word of mouth. Further attempts to distribute the software are currently in process. Ideally, the OffKey community will grow, and more people can enjoy learning to play a musical instrument while offering criticism to improve the software.

7. REFERENCES

[1] Dance Dance Revolution. Osaka, JP: Konami (1998).

[2] Guitar Hero. SunnyVale, CA: RedOctane (2005).

[3] Jaeschke, Rick. Piano Wizard. Music Educators Journal, 93 (4). 23.

[4] Java Sound API, 2008. Retrieved January 6, 2008 from Sun Microsystems: http://java.sun.com/products/java-media/sound

[5] Music Thought to Enhance Intelligence, Mental Health, And Immune System, June 22, 2006. Retrieved January 2, 2008, from ScienceDaily: http://www.sciencedaily.com/releases/2006/06/060622172738.htm

[6] Nguyen Duy. DashWork Studio, November 21, 2007. Retrieved January 6, 2008 from http://www.dashworkstudio.com

[7] Pfisterer, M. and Bomers, F. Java Sound Resources, February 17, 2005. Retrieved January 6, 2008 from http://jsresources.org

[8] Piano Wizard, (N.D.). Retrieved January 6, 2008, from Allegro Multimedia Inc.: http://www.pianowizard.com

[9] Rebetez, C. and Bentrancourt, M. Video Game Research in Cognitive and Educational Sciences. Cognitie, Creier, Comportament/Cognition, Brain, Behavior. 11 (1). 131-142

[10] Software Midi Keyboard Lite, 2006. Retrieved January 17, 2008 from Midi Masters Association: http://www.midimass.com/page.php?name=software_midi_keyboard_lite

[11] Stewart, B. Ezell, S., DeMartino, D., Fifai, R., and Gatterson, B. Virtual Technology and Education. Quarterly Review of Distance Education. 7 (4). 377-385