Themed Learning with Music and Technology

Debra L. Smarkusky
dls102@psu.edu
Information Sciences and Technology

Sharon A. Toman
sat11@psu.edu
Music Department

Penn State University
Dunmore, PA 18512, USA

Abstract

Interdisciplinary teaching and interpretation of technology for various applications provides a challenging and engaging environment for students to enhance their creativity, critical thinking and problem-solving skills. This paper presents results of a joint effort between faculty in the Department of Information Sciences and Technology and the Department of Music to develop projects at various levels of the curriculum that utilizes a themed learning approach to enhance learned concepts. Students utilize technology as a means to convey their creativity, artistic design, and appreciation of the Arts. Since the creation of digital musical scores, animations and software applications share a similar development process, active-learning exercises using a variety of technologies, provide students with an opportunity to enhance learned concepts in both disciplines. While experiencing the project development process, students are learning object-oriented terminology, animation frameworks, computer programming, distributed computing concepts, and principles of music theory.

Keywords: Interdisciplinary Projects, Java, Music, LEGO MINDSTORMS NXT, Sibelius, Flash, Alice

1. MOTIVATION AND RELATED WORK

Because many of the most pressing problems of the day are best solved using interdisciplinary approaches, it is important that students be well educated in their own disciplines and at the same time be prepared to engage in interdisciplinary projects (Amoussou, Boylan, Peckham, 2010). The coalescence of arts and technology allows innovative research opportunities while upholding the essential principles of general arts education (Izmirli & Baird, 2002). Within academia, faculty are seeking to include interdisciplinary projects with challenging computational components to enhance collaboration between computer science and other disciplines such as engineering, physics, chemistry, arts, and the social sciences. Building bridges between academic areas encourages students to explore, build hypotheses, experiment, development critical thinking for problem solving (Barr, Liew, & Salter, 2010).

Music compositions and software applications share an analogous project development process. Because software processes are complex and there is no ideal process, many organizations tailor the fundamental activities to match their needs while enhancing the quality of software and performance of developers (Sommerville, 2016). While musicians use music notation to design and interpret musical concepts, software developers use designs to convey architecture and algorithmic processes for programs. Similar to the object-oriented paradigm in software development, design patterns were compared to the various conventions of music notation and how to incorporate them into the design of music

www.manaraa.com

notation software (Brandorff, Lindholm & Christensen, 2005). An interdisciplinary music and animation course with various technologies for object-oriented programming, music notation and animation presented a project development process (requirements, design, implementation and test/debug phases) that exists between disciplines (Smarkusky & Toman, 2009).

Computing technology can be utilized to enhance our creativity within the learning process. Students should have opportunities to engage in interdisciplinary projects where they can build confidence and enhance their learning. Barker completed an interview study of both faculty research mentors and undergraduates that analyzed the organizational, social and intellectual conditions under which undergraduate research was being conducted (Barker, 2009). Barker observed that undergraduate research projects in science, technology, engineering, and mathematics (STEM) have benefits such as "improved retention in both the major and discipline-related careers; ability to work independently and to communicate well with a team; increased confidence in academic knowledge and technical skill; broader awareness of the discipline; and awareness of career opportunities and support for making career choices" (Barker, 2009).

Replacing the traditional introductory computer science course in the curriculum with an interdisciplinary and "connected" pair of courses in creative arts, humanities, history, math and computer science, natural sciences, and social sciences has resulted in an increase in female enrollments, retention in computing, and new energy for interdisciplinary research opportunities. Connecting with Art courses in the curriculum includes design (creativity) and development (computational thinking) (LeBlanc, Armstrong, & Gousie, 2010). An interdisciplinary project that requires computer science and music majors to develop a notation for playing a non-musical instrument resulted in better appreciation of the importance of musical notation and software syntax in both disciplines (Heines, Greher & Kuhn, 2009).

Students majoring in computer science, mechanical engineering, interactive multimedia and music designed and developed "Conducting Robots", where a robotic and graphical conducting system direct an orchestra and provided an opportunity for students to focus on critical thinking, creative problem solving, and computational thinking skills (Salgian, Ault, Nakra, Wang & Stone, 2011). Glasser asserted that: being active while learning is better than being inactive. Most people learn only 10% of what they read, 20% of what they hear, 30% of what they see, 50% of what they see and hear, 70% of what they talk over with others, 80% of what they use and do in real life and 95% when they teach someone else (Glasser, 1998).

In the remaining sections of this paper, we specify the technologies that were selected for the completion of our interdisciplinary projects which include Alice, Flash, Sibelius, LEGO® MINDSTORMS® NXT, and leJOS. We then provide a summary of the projects that utilize these technologies and discuss the scaffolding of knowledge and support for enhancing learned concepts. We provide student comments about these projects and conclude with student feedback and summary.

2. MUSIC/ANIMATION TECHNOLOGIES

Alice

Alice (www.alice.org) is a freely available programming environment for the teaching of object-oriented concepts in introductory programming languages that was created and distributed by Carnegie Mellon University (Cooper, Dann, & Pausch, 2000). The Alice interface, shown in Figure 1, provides informative

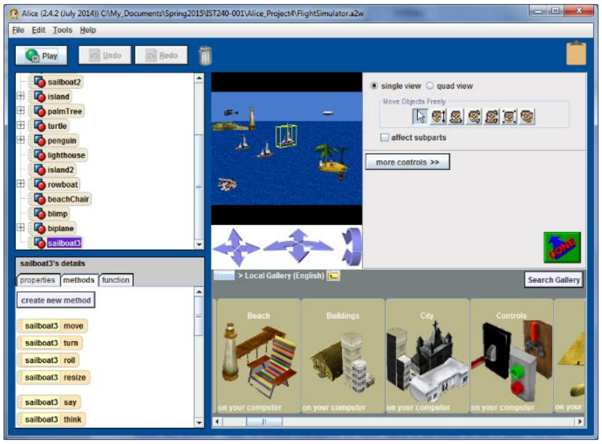

Figure 1. The Alice Interface

tutorials, a library of graphical components with embedded properties, methods and properties, camera controls, action events, and the ability to incorporate sound files all within an easy-to-learn drag-and-drop interface for the development of animations.

www.manaraa.com

Flash
Adobe Flash Professional (www.adobe.com) is a development environment for creating animations and multimedia content. This platform provides students with the ability to develop interactive media that focuses on design aspects such as scenes, frames, and timelines and incorporates vector graphics, audio, images and scripting during development of media-enhanced projects. An example of the Flash Interface is shown in Figure 2. This tool provides

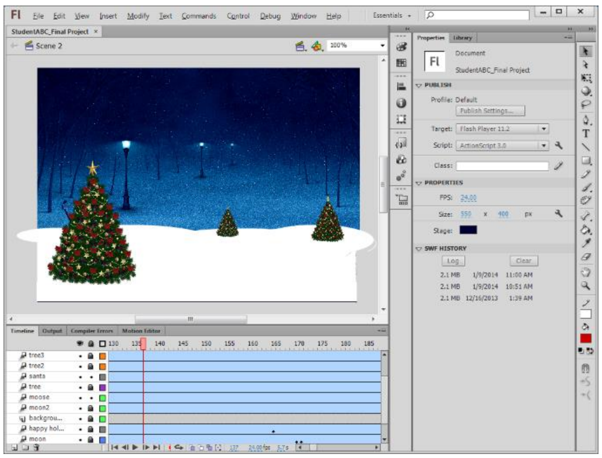

Figure 2. Flash Developer Interface

students with the ability to enhance their design experience via graphic layers and frameworks to better prepare them for the design and development of Java FX projects.

Sibelius
Sibelius (www.avid.com/US/products/sibelius**)** is a music notation application that is used by composers, arrangers, publishers, educators, and students to compose and write music. Sibelius, shown in Figure 3, provides users with a click-and-drag format for the insertion of musical notes and allows for instant playback of the musical score. This software application contains editing

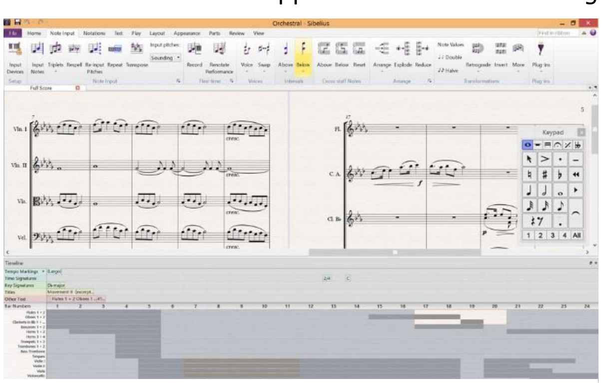

Figure 3. Sibelius Music Notation Application

tools for the management of notes, insertion of text and graphics, selection of musical instruments, and utilization of an assortment of built-in sound plug-ins.

LEGO MINDSTORMS NXT
The LEGO® MINDSTORMS® NXT platform, shown in Figure 4, provides the framework for students to experience software development using sensor input, multiple software threads of computation, and wireless communication. The

Figure 4. LEGO® MINDSTORMS® NXT

NXT brick includes four sensor ports ('1', '2', '3', and '4') and three input ports ('A', 'B', and 'C'). LEGO offers a wide variety of sensors to include a Sound Sensor, Light Sensor, NXT Color Sensor and Ultrasonic Sensor. Communication between the computer and the NXT brick can be realized with a USB cable connection or programmed to use the Bluetooth**®** (www.bluetooth.com) wireless communications. We elected to utilize this technology due to the abundance of available documentation, less expensive cost of the robot kits, and the extensible nature of the platform for various projects (LEGO MINDSTORMS Education NXT, 2015).

LeJOS NXJ
Originally created from the TinyVM project, which was an implementation of a Java Virtual Machine for LEGO® MINDSTORMS® RCX system, leJOS NXJ is a Java programming environment for the NXT (leJOS Team, 2009). This firmware is used to replace the NXT factory-loaded software, which includes a drag-n-drop environment for programming. This leJOS NXJ environment includes a Java Virtual Machine, a library of classes that implement the NXJ Application Programming Interface (API) for execution on the brick, a library of Java classes for computer programs that communicate with the brick via USB or Bluetooth, PC tools for debugging and

www.manaraa.com

flashing the firmware, and the capability to compile, link and upload programs and other files to the NXT brick. The benefits of using leJOS NXJ was that it utilizes the Java programming language and can be developed using the NetBeans (www.netbeans.org) or Eclipse (www.eclipse.org) Integrated Development Environment (IDE) with associated plugins for both environments. Students enhance their problem-solving skills using object-oriented open source software to create challenging multimedia-based solutions.

3. INTERDISCIPLINARY PROJECTS

The content contained in our music and animation projects are presented in a scaffolding approach and utilize the seven principles for good practice in undergraduate education as presented by Chickering and Gamson as shown in Table 1 (Chickering & Gamson, 1987).

Encourages contacts between students and faculty.
Develops reciprocity and cooperation among students.
Uses active learning techniques.
Gives prompt feedback.
Emphasizes time on task.
Communicates high expectations.
Respects diverse talents and ways of learning.

Table 1. Principles for Undergraduate Education

Knowledge learned in earlier projects can be directly applied to concepts needed for future projects. Learning starts at the beginning of the curriculum when the basic animation and object-oriented concepts are introduced using Alice. This learning continues with advanced design and animation concepts in Flash and the composition of digital music scores in Sibelius. This knowledge provides the foundation for the design and development of more advanced interdisciplinary projects using Java via LEGO® MINDSTORMS® NXT platform and leJOS. All of these projects provide students with a better appreciation of the shared project development process that exists between programming, animation and music. In addition, these projects provide collaborative research and learning opportunities between disciplines, inspire ideas for advanced computing projects using multimedia, and can be utilized as a platform for interdisciplinary projects between technical and general education courses.

Introductory Java Course
Students in the Information Sciences and Technology (IST) program first experience interdisciplinary projects in the Introductory Programming Course, which can be completed in the first or second semester of the curriculum. Students are learning the object-oriented terminology for the first time and Alice provides a fun environment to familiarize students with these concepts. Students are introduced to the Alice World and the associated gallery that contains classifications for the types of objects that can be added to a scene. Once objects are instantiated (added to the scene), we discuss the properties (attributes) and behaviors (methods) for each object shown in the Object Tree. Students also learn how to add sound effects and background music to enhance their animations. Each project provides students with an opportunity to be creative while learning the importance of organizing concepts and behaviors. The creation of new methods allows students to reuse the code they had previously created in other areas of their animation, especially event-handling. Logical structures provide variable behaviors, while event-handling and camera controls provide a more interactive experience.

Alice provides the perfect learning environment for which requirements definition, design, implementation and test/debug activities could be introduced with instant visual feedback about their performance. The project description represents the requirements definition. Students identify various classifications from the gallery and determine how they would be used in the animation. For the implementation phase, students add instances of the objects to the world, position them as specified in their design, and utilize the drag-and-drop interface of Alice to create new methods, events and logical structures to realize the required behaviors and actions of their animation. Students select the play button to ensure correctness and completeness and make changes to the logic in methods or placement of the object within the scene. The Alice environment provides a good introduction to object-oriented terminology, software development phases and the utilization of sound files for special effects. Once students complete several projects using Alice, they are introduced to the Java programming language.

General Education Arts Course
The INART236 - Introduction to Music and Animation with Technology course is a General Education Arts course that students are encouraged to complete after the second semester. The fundamental guidelines for a

www.manaraa.com

General Education course at our university states that the course "aids students in developing intellectual curiosity, strengthened ability to think, and a deeper sense of aesthetic appreciation" (Penn State University, 2012). In this course, Sibelius is utilized to aid in learning of music theory and the creation of a musical score, Flash allows students to realize more complex animation concepts, and a capstone project at the end of the course combines lessons learned in both disciplines (Smarkusky & Toman, 2009).

The first six weeks of the semester focuses on the basic fundamental concepts of music theory and composition using Sibelius. Learning the rudiments of music requires an understanding of each of the elements of music before they can sing, play or write music notation. The fundamental concepts of music theory is comprised of notation, scales, meter, rhythm, intervals; and basic chord structure. As students are learning each of these music fundamentals they are working with technology via the interface of Sibelius to solve problems through active-learning exercises. These exercises, which are created in Sibelius, challenge the student's ability and knowledge about each learned concept. The click-and-drag interface of Sibelius allows students to easily insert a musical note and its rhythmic value into a digital music score. In addition to learning concepts of music theory and composition, students gain knowledge about a wide variety of musical instruments that include strings, woodwinds, brass, percussion, and keyboard instruments. This knowledge provides students with the needed understanding to allow them to incorporate staves for each of the newly added musical instruments during composition. In addition, Sibelius provides additional plug-ins for adding drum patterns, guitars, rhythmic patterns, etc. The process of creating a musical composition from scratch is a time consuming process of adding notes for a simple melody, chords for harmony, articulation and expressive nuances as well as instruments. It is a step-by-step process that must be followed correctly for a successful digital musical score. Music has often been compared to learning a foreign language. Students must learn and have an understanding of the basic music theory terminology to be able to understand the more complex components required to complete a digital musical score.

Students then learn to develop more complex animations using Flash during the second six weeks of the course. The Flash framework provide students with an opportunity to design animations that include timelines, scenes, layers, and frames. Gridlines, rulers and guides provide guidance for the drawing of objects, creation of symbols or text components. Students learn how to create frame-by-frame animations, motion and shape tweens, and graphic symbol animations. The more advanced features include animated text, motion guide layers, mask layers, and the inclusion of sound. Interactive features include frame labels and buttons with ActionScript being utilized for more complex actions. Once again, students are gaining experience with the project development process (Smarkusky & Toman, 2009). To ensure that students are gaining experience using a specific animation component, project descriptions and expected results are provided for each exercise. During the design phase of the project, students need to identify various scenes, determine layers, and the animated effects that would occur at each frame of the timeline. Similar to creating graphical user interfaces for their software development courses, students need to focus on the design of their interface (scene) and layer components to achieve the desired result. Students design in a top-down approach and utilize a bottom-up approach for implementation. Each animation component is added to a layer, layers are added to scenes, and scenes are integrated to create the final animation. Students play the animation to check that all requirements were satisfied and make appropriate modifications when needed.

During the last three weeks of the course, students are assigned a capstone project that focuses on combining knowledge gained in both Sibelius and Flash. The goal of the project is to create an animated greeting with background music. For the music component of the project, the requirements definition is a simple and introductory piece of sheet music, which is provided to students for them to compose digitally using Sibelius. Students need to pay specific attention to detail to ensure that key signatures, voices, notes, accidentals, articulation, etc. match those in the requirements definition. Students are allowed to enhance the musical piece by adding additional staves for various instruments as well as utilize a software plug-in to create a percussion part to enhance the style of their composition. Once testing is completed, students are required to export the musical score and convert the file to an MP3 format for use in Flash.

For the animation component, students must satisfy the specified technical components of the animation while keeping within the theme of the assigned musical song. Technical requirements include the addition of a separate layer for

www.manaraa.com

incorporating the sound file that was created in Sibelius, creation of a new symbol, animated text block with individual letter animation, use of a mask layer, motion tweens, shape tweens, action frames and labels, and button implementation to include ActionScript. The open-ended nature of the animation allows students to gain experience in requirements definition and apply knowledge gained from active-learning and homework exercises to the design phase of the animation. Implementation and testing is an iterative process as students need to ensure the frame-by-frame execution of the project is correct to that point of completion. The assessment for this project is based on the correctness and completeness of their musical composition, their creativity in correlating the animation with the theme of the music, and the correct implementation of the technical components of the animation.

Advanced Java Course
During their junior year, IST students are required to complete a Distributed Computing course. One of the projects within this course is the LEGO Maze Project as demonstrated in Figure 5 (Smarkusky, Toman, Sutor & Hunt, 2013). Students gain experience with the leJOS interface

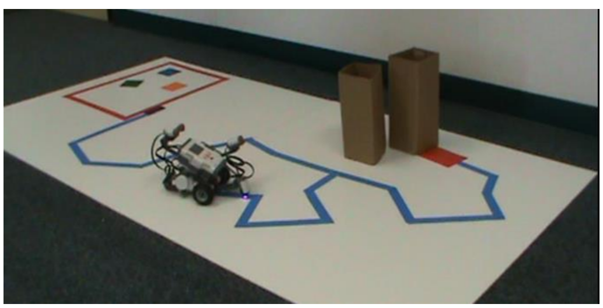

Figure 5. LEGO® Maze Project

using Java where they can compile, upload, and execute programs on the LEGO® MINDSTORMS® NXT. Unlike traditional software projects that obtain input from graphical user interfaces, files or databases, this project utilizes input from the sound, color and ultrasonic sensors of the robot. The values read from the sound sensors are impacted by environmental acoustics; robot motors react differently based on the flooring surface; color sensor values change with sunlight, shadows, natural or florescent lighting; and ultrasonic senor values are constantly changing due to range of vision and constant movement of the robot.

This project requires that students utilize user defined sound files or the onboard sound clips that are provided with the leJOS platform. In preparation for a future project, students must use the sound sensor to start the robot in motion (e.g. a hand clap that is at least 25 decibels). Once in motion, the robot must follow a specified path (Color1). Upon reaching the first wall, the robot should turn right 90 degrees and continue following the path. When it reaches the second wall with a different color (Color2), it will turn 180 degrees and continue following the Color1 path backwards, keeping on a straight path when possible. Upon reaching the sand box (Color3), the robot will turn left 90 degrees and begin a search pattern. The goal within the sandbox is to find three additional color swatches (Color4, Color5, Color6), and play a unique sound for each of the identified colors. The challenging nature of this project provides an opportunity for problem-solving, learning the LeJOS Application Programming Interface (API), and gaining additional experience with Java Threads.

Once familiar with the leJOS API, sensors, and the ability to maneuver the robot, IST students are assigned a project in which they must work with students from a General Education Music Theory course to create a robot dance where the movements of robot couples (two robots) are synchronized to the rhythm of the music (Smarkusky & Toman, 2013). With the choreography of the dance being the bridge between disciplines, the background music is created from scratch by the music students and the robot dance movements are implemented by the IST students.

Within the music theory course, students are introduced to more complex characteristics of melody, rhythm, harmony, and instrumentation. Similar to the General Education Arts course, students start with a more advanced piano musical score that is then created digitally using Sibelius. This project is further enhanced using various instrumental parts to include flute, trumpet, saxophone, guitar, bass guitar, and drum pattern plug-ins. Students are able to control tempo, dynamics, and other music nuances using the Sibelius's integrated *Playback* feature.

The IST students implement Java applications, utilizing the leJOS NXJ environment, to maneuver the robots based on the steps identified in the choreography which was also designed by the music students. This provides unique learning opportunity for the IST students as they realize the importance of threads and the completion of object-oriented methods for each routine with parameters for time, duration, direction, or rotation for various movements. The timeline

www.manaraa.com

that is included with the *Playback* feature of Sibelius provided assistance with synchronization of the robot dance movements to the music file. Assessment for this project is based on the correctness and completeness of the technical requirements of the project for each discipline along with the complexity of the choreography and how well the movements were synchronized to the music. Similar to the comparison presented by Do & Gross (Do & Gross, 2007), the programmer is not expected to be a music student, and the music student is not expected to be a programmer. Each member of the interdisciplinary team utilizes their strengths for the success of the project.

4. FEEDBACK AND SUMMARY

Overall, the results of incorporating a musical theme to allow students to learn technical concepts was a positive experience. At the freshman level, Alice provided a platform to excite students about music and animation, the project development process, and object-oriented terminology, as a preamble to the development of projects using Java. During the sophomore year, students were encouraged to enroll in the General Education Arts course in music and animation. Since the development of digital musical scores, animations, and software applications follow a similar project development process, the completion of this course provided students with opportunities to understand the importance of paying attention to details (within Sibelius and Flash), learning how the project development process can be applied to other disciplines, and providing students with an opportunity to learn more about designing interfaces using layers that will provide a foundation for the learning Java FX in future Java programming courses.

The General Arts Education course was offered in the Fall 2013 semester. This course had an enrollment of 20 with 14 students electing to complete an optional survey. As shown in Table 2 of the Appendix, and represented below with actual student comments, the students enjoyed using various technologies, had a better understanding of the project development process, increased their creative abilities, gained confidence with the application of learned concepts to other disciplines, and had an increased appreciation for the attention to detail and time involved with creating musical animations.

> *"Completing my music and animation project did change my perception of music, animation, and technology. It was pretty technical to make pieces of music or animation, yet putting it all together was a little difficult, yet challenging and fun."*

> *"When I look at music and animation, I see how much work goes into it and I enjoy it much more. I now see the bigger picture and see that you can get very creative with it if they are willing to learn."*

> *"Music is good for teaching logic and animation can teach patience. Also, they both can give students an edge with some jobs."*

> *"I think the arts are an underappreciated aspect of our culture and music and animation go hand in hand today. You don't have to go far to see some great achievements online using a combination of these. I would definitely say that getting an introduction into the complexity to gain appreciation for it is a good idea."*

Since students required more experience with Java to include concurrent threads of control and programming with limited resources, we elected to wait until the advanced Java course for the application of projects using LEGO® MINDSTORMS® NXT and the leJOS firmware. Assigning the LEGO Maze project as required project in the course, allowed students to gain experience and confidence with these new technologies prior to being paired with the music theory students for the completion of the LEGO Dance project. Using the leJOS programming to develop programs for the NXT can challenge one's software development, robotics, and programming skills. The LEGO Dance project incorporated a new perspective to the software development process in that students now needed to implement the requirements (choreography) that were designed by the music theory students, and be able to explain the limitations of the technology and the project (keeping two robots on a 6'x12' floor surface, implementing moves via two robot wheels and not actual human feet, etc.). The expectations of both the IST and music students were high, so they worked together as a team to complete and solve the challenges of this project. At the end of the project, we surveyed all students, both music and IST, to get their feedback. We have incorporated the LEGO Maze and LEGO Dance Project into the advanced Java course every spring semester since 2012. The combined feedback for the LEGO Dance Project over the past four years with 96 students participating is

www.manaraa.com

shown in Table 3 of the Appendix, and includes responses from both Music and IST students.

The results demonstrate that the LEGO Dance Project was a success. Students were able to express their creativity, enjoyed working with students from other disciplines, learned more about animation and recommended that this project be offered again in the future. The project teams consisted of three music students for every one IST student. We expected the response rate for learning more about music composition to be lower than other results. This is based on the fact that the music students were involved with both music and animation components of the project, as compared with the IST student who participated only in the animation component of the project and did not have any involvement with music composition. Some additional student comments about the LEGO Dance Project are shown below:

> *"It was cool to be able to make robots dance to the music we created, I never even would have thought of incorporating IT with music. I enjoyed the accomplished feeling at the end of the day to be able to put all of this together and make it work."*

> *"Being able to animate the robot was fun, but the limited physical abilities of the robot made it challenging and frustrating. Incorporating creativity in the project was accomplished when, as a group, we all discussed the limitations and had to creatively find a solution around these limitations that still looked appealing."*

> *"I think this project is a great way to test the knowledge of students and work in teams. It was a good experience meeting with IST students and choreograph the dance."*

> *"Working with the robots as opposed to the same IDE, GUI, and box is fantastic. It's a wonderful experience to have a "business" that gives "demands" that we need to deliver. Especially so when they don't 100% understand the constraints of the code and language so we need to act as intermediaries. This was an outside the box project and I truly enjoyed it."*

In summary, these interdisciplinary projects provided students in both Music and IST courses with an opportunity to become familiar with terminology and concepts from both disciplines. Students quickly learned that the creation of digital musical scores, animations, and the Java programs that controlled the choreography all required attention to specific detail and the following of a very similar development process. In identifying tools for application of concepts, both departments preferred software that would provide instant feedback to students during project development.

By using Alice, Flash and Sibelius, we found that students are trained to look for specific details that are required for the creation of a correct and complete deliverable in both music and animation. This attention to detail and shared project development process provides additional experiences to students for the development of Java applications and the utilization of leJOS NXJ firmware for the LEGO maze and LEGO Dance projects. The success of these projects and support from student feedback warrants the continued offering of these projects in the future. In addition, we have been asked to offer the General Education Arts course as part of a new honors program at the campus. Students who have participated in these projects requested that faculty offer undergraduate research opportunities in music and animation. We will continue utilizing the LEGO® MINDSTORMS® NXT and the leJOS firmware for current projects and plan to incorporate the new LEGO® MINDSTORMS® EV3 into additional undergraduate projects in the next academic year.

5. REFERENCES

Amoussou, G., Boylan, M., & Peckham, J. (2010). Interdisciplinary computing education for the challenges of the future. In *Proceedings of the 41st ACM technical symposium on Computer science education* (SIGCSE '10). ACM, New York, NY, USA, 556-557. DOI=http://doi.acm.org/10.1145/1734263.1734449

Barker, L. (2009). Student and Faculty Perceptions of Undergraduate Research Experiences in Computing. *Transactions on Computing Education* 9(1), Article 5. March 2009, 28 pages. DOI=10.1145/1513593.1513598 http://doi.acm.org/10.1145/1513593.1513598

Barr, V., Liew, C.W. & Salter, R. (2010). Building bridges to other departments: three strategies. In *Proc. of the 41st ACM technical symposium on Computer science education* (SIGCSE '10). ACM, New York, NY, USA, 64-65. DOI=10.1145/1734263.1734285

www.manaraa.com

http://doi.acm.org/10.1145/1734263.1734285

Brandorff, S., Lindholm. M. & Christensen, H.B. (2005). "A Tutorial on Design Patterns for Music Notation Software", *Computer Music Journal*, 29(3), pp. 42–54.

Chickering, A. W. & Gamson, Z. F. (1987, March). Seven Principles for Good Practices in Undergraduate Education. *The AAHE Bulletin*. Retrieved on July 1, 2015 from http://www.aahea.org/aahea/articles/sevenprinciples1987.htm

Cooper, S., Dann, W. & Pausch, R. (2000). "Alice: A 3-D Tool for Introductory Programming Concepts", *Journal of Computing Sciences in Colleges*, 15(5):108-117.

Do, E.Y. & Gross, M.D. (2007). Environments for creativity: a lab for making things. In *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition* (C&C '07). ACM, New York, NY, USA, 27-36. DOI= http://doi.acm.org/10.1145/1254960.1254965

Glasser, W. (1998). *Choice Theory in the Classroom*, New York: Harper Collins, ISBN: 0-06-095287-3.

Heines, J., Greher, G. & Kuhn, S. (2009). "Music Performatics: Interdisciplinary Interaction", *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE),* pp. 478-482.

Izmirli, O. & Baird, B. (2002, May). "A Model for Integrating Arts and Technology in a Liberal Arts College", *Journal of Computing Sciences in Colleges*, 17(6), pp. 102-109.

LeBlanc, M.D., Armstrong, T. & Gousie. M.B. (2010). Connecting across campus. In *Proceedings of the 41st ACM technical symposium on Computer science education* (SIGCSE '10), ACM, New York, NY, USA, 52-56. DOI=10.1145/1734263.1734280 http://doi.acm.org/10.1145/1734263.1734280

LEGO MINDSTORMS Education NXT (2015). https://shop.education.lego.com/legoed/en-US/search/navSearchResults.jsp?categoryId=EDU_PRD_LINE_106&ProductLine=MINDSTORMS+Education+NXT, accessed July 2015

leJOS Team (2009). http://lejos.sourceforge.net/

Penn State University (2012). University Faculty Senate, Policies and Rules for Undergraduate Students, Baccalaureate Degree Curriculum, http://senate.psu.edu/policies-and-rules-for-undergraduate-students/appendix-a-1-general-education-baccalaureate-degree/, accessed July 2015.

Salgian, A., Ault, C., Nakra, T.M., Wang, Y., & Stone, M.K. (2011). Multidisciplinary computer science through conducting robots. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (SIGCSE '11). ACM, New York, NY, USA, 219-224. DOI=10.1145/1953163.1953229 http://doi.acm.org/10.1145/1953163.1953229

Smarkusky, D.L. & Toman, S.A. (2009). An interdisciplinary approach in applying fundamental concepts. In *Proceedings of the 10th ACM conference on SIG-information technology education* (SIGITE '09). ACM, New York, NY, USA, 224-228. DOI= http://doi.acm.org/10.1145/1631728.1631800

Smarkusky, D.L. & Toman, S.A. (2013). An Interdisciplinary Learning Experience: The Creation of a Robot Dance, *Information Systems Education Journal* 11, 5, pp 55-62.

Smarkusky, D.L., Toman, S.A., Sutor, Jr., P. & Hunt C. (2013). Performing robots: innovative interdisciplinary projects. In *Proceedings of the 14th annual ACM SIGITE conference on Information technology education* (SIGITE '13). ACM, New York, NY, USA, 125-130. DOI=10.1145/2512276.2512285 http://doi.acm.org/10.1145/2512276.2512285

Sommerville, I. (2016). *Software Engineering*, Tenth Edition, Pearson, ISBN: 978-0-13-394303-0.

www.manaraa.com

APPENDIX

Question (1=Strongly Disagree; 5=Strongly Agree)	Mean	Std Dev	Positive Responses (%)
I now understand the fundamental concepts of music theory and the technical components of animation for the creation of a musical animation	4.21	0.58	93
I enjoy using various technologies for the development of digital music scores and animations	4.21	0.58	93
I now understand the phases of the project development process that are necessary for the creation of digital musical scores and animations	4.29	0.61	93
During this course, I developed and applied skills in creative thinking through design and active learning exercises	4.29	0.47	100
I enjoyed creating a musical animation that encompassed all learned concepts, that provided an opportunity for the expression of creativity as well as interdisciplinary and integrated learning	4.07	0.73	79
I feel that the information learned in this course could enhance projects completed in other disciplines	4.14	0.53	93

Table 2. General Education Arts Course Feedback

Question (1=Strongly Disagree; 5=Strongly Agree)	Mean	Std Dev	Positive Responses (%)
This project was a creative learning experience	4.28	0.68	92
I enjoyed working with students in other disciplines	4.25	0.80	89
Both Music and IST student worked together as a team to create a successful and complete project	4.33	0.74	94
I learned more about music composition during this project	3.68	0.95	63
I learned more about animation during this project	4.19	0.81	84
I would recommend offering this project again to Music/IST students in the future	4.21	0.83	85

Table 3. LEGO Dance Project Feedback

المنارة للاستشارات
www.manaraa.com