ISSN: 2167-1435
v29 n1937

Bridging Game-Programming into theK-12 Curriculum

Shannon, Li-Jen
lys001@shsu.edu
Computer Science Department
Sam Houston State University
Huntsville, TX 77340 USA

Dalat Ward, Yaprak
y_dalatward@fhsu.edu
Advanced Education Programs
Fort Hays State University,
Hays, KS 67601 USA

Abstract

The fact that the U.S. students lag behind in math and science performances continues to be a burning issue for the nation. In the past decade, although ample studies offered a significant variety of fun computing projects to motivate students in math and science learnings, this issue has been disregarded perhaps due to misaligned curriculum, or due to perhaps educators who lacked the required technical skills. To remedy this problem, this study investigated how the perspectives of the non-computer science educators changed after learning game-programming and how it could be fitted into the K-12 curriculum. Fourteen non-computer science educators and/or administrators in the K – 16 educational systems who made up a cohort at Sam Houston State University, Master of Education/Instructional Technology Program participated in this study. The participants were required to learn two free Web 2.0 game-programming applications and reflect on an article related to reviving interest in math and science as part of their program. Qualitative data consisted of online reflections, and peer-review processes through Facebook. A quantitative component was added to the analysis. The findings indicated that: (a) the perspectives of the participants changed from negative to positive as they reflected on their own game-programming learning experiences; (b) participants came to understand how game programming could build up students' logical concepts and critical thinking skills improving performances in math, science, and other subjects; and (c) due to the benefits of logical concepts and critical thinking skills game programming could have immense benefits if built into the K-12 curriculum.

Keywords: Scratch, Alice, Game-Programming, Project-Based Learning, Critical Thinking Skills

1. INTRODUCTION

In the past decades, the U.S. students have continued to fall behind in their math and science performances when compared to other nation students (Education Week, 2012, PISA, 2009). The students from Grade 8-12 seemed to lose their motivation in the science, technology, engineering, and mathematics (STEM) fields (Brown & Brown, 2009). To address this issue, many grants have been allocated to conduct research; and many grant projects have been developed to investigate how the educational systems could be modified to motivate student learning in STEM (PLTW, 2012; RAFT, 2012;

ISSN: 2167-1435
v29 n1937

Redmond, Thomas, High, Scott, Jordan, & Dockers, 2011).

Although the issue seemed bleak, the enrollment of computer science students increased when programs and concentrations in game development were added to the curriculum (Lewis, 2009). However, ever since the public acquired a misconception of younger generations being savvy in technology, the technology courses continued to be eliminated as core courses. Table 1 shows the curriculum structure changes concerning technology implementation as part of Texas high school graduation requirements in the past decades. It is disheartening to observe how the technology courses have been gradually eliminated from the curricula. In fact, this misconception has affected higher education as well. Reviewing the Texas Higher Education Coordinating Board's reforming core curriculum criteria, the required credits to fulfill a baccalaureate degree have been decreased in the past decade down to 42 credit hours (Texas Higher Education Coordinating Board, 2012). As the introduction to computer and the foundation of science courses have been gradually taken out from the core curriculum, educators started to witness the long term impact this decision had on the performances of U.S. students. Student performances in technology and science are all time low when looked at it from a global aspect (Hira, 2009) as well. Even worse, the educators' negative perceptions about math, science, and programming had directly impacted the students' interests and motivation in discovering the STEM fields.

The purpose of this research was to remove the fear and/or misconceptions about programming for the non-computer science educators. Many educators have the incorrect notion about the job descriptions and tasks in the technology and computer science fields. Moreover, many educators doubt that it is feasible or doable to learn game programming much less to teach game-programming in the K-12 curriculum. This study was designed to explore the feasibility of teaching game-programming. The research question which guided this study was: can game-programming bridge the students' learning gaps in all subjects?

Table 1

Texas State High School Graduation Requirements

Required Course	1997-2002 Credits	2004-2005 Credits	2007 – 2008 Credits	2012-2013 Credits
English	4	4	4	4
Mathematics	3	3	3	3
Science	2	2	2	2
Social Studies	2.5	2.5	2.5	3
Economics	0.5	0.5	0.5	N/A
Academic elective	1	1	1	1
Physical Education	1.5	1	1	1
Health Education	0.5	N/A	N/A	N/A
Speech	0.5	0.5	0.5	0.5
Technology Applications	1	N/A	N/A	N/A
Fine Art	N/A	N/A	1	1
Elective Courses	5.5	7.5	6.5	6.5
Total Credits	22	22	22	22

Note: Texas Education Agency, Chapter 74d, 74e, 74f, & 74g.

The following sections include literature review, methodology, findings, and finally conclusion to support this study.

2. LITERATURE REVIEW

STEM Development in the U.S.

Since the globalization of the job market, the U.S. STEM workers have been faced with a myriad of challenges as many companies started to move their high-value operations to off shore. The U.S. corporate leaders agree that globalization has fundamentally changed the way in which they manage their human resources (Hira, 2009). Many companies hired off shore high-tech employees in order to reduce their Research and Development expenses. These practices resulted in growing concerns among U.S. authorities. As a result, an urgent action became imperative to nurture and encourage highly skilled U.S. workers.

From the Duke Today News, Lynn (2011) reported that U.S. students lag behind their peers throughout the world in science and math. To address this critical issue, a few Duke University faculties developed a program called "Project Lead the Way (PLTW)" to teach Alice to the teachers from middle schools and some teachers from high schools. The outcomes of this

 ISSN: 2167-1435
v29 n1937

program showed that through PLTW, with the real-world, problem-solving based curriculum, more than 1,000 students were able to benefit from the wide range of classes in STEM fields.

Programming Misbelief

Many students have a misconception or no perception of what the computer scientists do (Carter, 2006). Consequently, the students' disinterest in computer science is due to a lack of familiarity with the subject.

There is a common belief that computer science is an intimidating topic. Simha (2012) stated that "many people incorrectly believe that a CS career is all about programming" (p.1). In addition, Lewis (2009) shared the public views of the CS "discipline is more commonly associated with the image of an introvert who spends most of his time staring at computer screens rather than interacting with people" (p.1).

It is also noteworthy that Kelleher and Pausch (2007) reported how the freshmen who had declared their major in computer science dropped by 70 percent in the five years early in the century.

As multimedia development became more user-friendly, programming software became more advanced, and created a great playground for the educational landscape. Baldwin (2007) stated that computer programming requires an aptitude for solving problems using different types of software so the users are capable of learning new areas while making trial and error practices.

The concepts of any programming software are based on understanding how to make different pieces work together in order for the animation to run smoothly and without errors. Scratch and Alice are the two most popular programming freeware which enable users to download the system directly and share with their peers. These two programming freeware allow users to immediately see how the programs run to understand relationships between programming statements and respective movements of objects that create animation. The game programming Alice for an example provides users with tools to learn fundamental programming concepts (Giansante, 2009). In fact, Chris Betcher, in his blog "Teaching Kids To Think Using Scratch" defined the "big ideas of programming" as the process of learning problem solving, thinking mathematically, and using logic reasoning (Betcher, 2010). Oddie, Hazlewood, Blakeway, and Whirtifle (2010) also agreed that problem solving and critical thinking are precursors to any programming activity.

Logical Thinking

Sezen and Bulbul (2011) defined logic as "the discipline that examines the structure of knowledge and distinguishes correct and wrong reasoning, is also known as the tool of correct thinking" (p. 2476). Lawson, Banks, and Ve Logvin (2007) emphasized that logical thinking ability is the primary factor that influenced both students' self-sufficiency and their achievement in science. After studied the science laboratory activities, Koray and Koksal (2009) concluded that "logical thinking as an aim of higher order is mirror of thought that come about in formation of operations in child" (p.2).

Critical Thinking

Living in this digitalized society, it is vital for all of the social network citizens to use their critical thinking skills verifying the free online information to make the best possible decision-making in their daily life. Feldman (2009) stated that "critical thinking involves evaluating a situation, problem, or argument and choosing a path of investigation that leads to the best possible answers" (p. 10). Cottrell (2011) described a well balanced person based being able to handle: (a) an argument, (b) clarity, consistency and structure of the argument, (c) flaw/s in the argument, and (d) sources of evidence. By working with interactive programming software, the users are practicing the process of critical thinking skills in order to have a successful outcome.

Scratch

Scratch is an interactive programming language which makes it easy to create interactive stories, animations, games, music, and art that can be shared on the web (MIT, 2012). Scratch is designed to offer a learning environment which includes developing problem-solving skills, learning how to think creatively, and reasoning systematically. With little or no experience in programming, users can learn and use Scratch quickly. Scratch is a two-dimensional programming software where the user interlocks blocks that represent different commands to move different "sprites" (objects) or characters (MIT, 2012; Utting, Cooper, Kolling, Maloney & Resnick, 2010). With this simple functionality, Scratch is a suitable first language for those who

 ISSN: 2167-1435
v29 n1937

are learning to program (Tangney, Oldham, Conneely, Barrett, & Lawlor, 2010).

The "sprites" and background needed in a game design can be imported from the Scratch database. Or users can draw, create, and download their own designs as the sprites and background. The Scratch Editor or any other image editors have the capabilityof creating new images for Scratch projects.

Scratch offers puzzle pieces or build-blocks capabilities to easily create interactive stories, animations, and games by adding scripts or snapping graphical programming blocks together while allowing for programmable manipulation of media (IDLBI, 2009). To create a visual layout of the game design, the blocks of codes can be printed separately as a picture in different windows with various function groups.

Furthermore, accompanied by a great capacity of being copied and of shared function blocks, collaboration, group work can become part of the learning task (Lamb & Johnson, 2011). Scratch allows automatic project uploads to the Scratch website allows projects to be embedded in any website for sharing. Sharing learning is accentuated by a user community where games and stories created in Scratch can be uploaded, downloaded, or just used for playing.

Alice

Dann and Cooper (2009) stated that Alice was developed from the idea of "head fake" which enables students to learn one thing while learning another. This environment brings students together from various disciplines who believe that they are creating a virtual game world, but in actuality are learning how to perform team work, how to respect each other, and how to tap into learning math, science, and other subjects (Dann & Cooper, 2009).

Alice is an innovative 3D programming environment which enables users to easily create an animation, an interactive game, or a video to share on the web (Alice, 2012; Utting, Cooper, Kolling, Maloney, & Resnick, 2010). Alice offers a quad view platform. Viewing from the top to place the object or to maneuver its placement can prove invaluable as the game is set to run and the characters are called to transition according to placement and relation to each other (Villaverde, Jeffery, & Pivkina, 2009). Alice contains pre-built objects, methods, control logic, and events to direct actions on the screen in order to narrate a story. A simple storyboard can be completed by manually creating the scenery, objects, and then inserting the control methods to animate the game (Baldwin, 2007).

Alice is a teaching tool designed to be a student's first exposure to object-oriented programming (Alice, 2012). It can be used for more advance programming students. However, it is said that Alice is a user-friendly software as users can easily utilize the drag and drop commands to move individual pieces of characters or objects (Carnegie Mellon University, 2012). Sykes (2007) stated that Alice has the capability to introduce people who would not think of themselves as programmers, or who have not had the opportunity to use 3D programming. The users are able to review the entire game design process by selectingan option of printing the code in the html form.

Project-Based Learning

Buck Institute for Education (2009) shared the findings in implementing Project-Based Learning (PBL) approach in which PBL can be "the catalyst for an engaging learning experience and create a context for a powerful learning community focused on achievement, self-mastery, and contribution to the community" (p.1).

The use of PBL is "not only a potentially effective instructional approach, but it is also an essential component of several current school reform models" (Ravitz, Mergendoller, & Markham, 2004, p.2). It is also noteworthy:

> A growing body of academic research supports the use of project-based learning in schools as a way to engage students, cut absenteeism, boost cooperative learning skills, and improve test scores. Those benefits are enhanced when technology is used in a meaningful way in the projects (Edutopia, 2012, p.1).

In 2011, the National Assessment of Educational Progress at Grade 8 reported that the students who did hands-on projects every day or almost every day scored higher on average than students whose teachers reported students who did hands-on projects in class less frequently (Science 2011).

In addition, PLTW offers a hands-on, activities, project, project-based comprehensive curriculum which is aligned with relevant national standards and is collaboratively developed by subject matter experts. Based on the PLTW alumni assessment (PLTW, 2012) report: (a) the PLTW graduates were 5 to 10 times more likely to pursue engineering and technology classes than other first-year college students; (b) the PLTW graduates had a GPA 0.21 points higher than the average GPA of all first-year college students; (c) the PLTW graduates were more engaged in schoolwork than non-PLTW students were; (d) the PLTW graduates outscored a random sample of other career/technical students in reading, mathematics, and in science; (e) 79 percent completed four years of college-preparatory mathematics and 63 percent completed four years of college-preparatory science; and (f) 97 percent said they planned to pursue a four-year degree as opposed to 67 percent of non-PLTW students.

3. METHODOLOGY

This study investigated how the perspectives of non-computer science educators changed in learning game-programming in the K-12 curriculum. The research question which guided this study was: can the game programming bridge the students' learning gaps in all subjects. Both qualitative and quantitative data analysis were used to first conduct qualitative analysis. Once the qualitative findings were defined, the data were quantified by converting textual constructs into numbers (Tashakkori and Teddlie, 2003). This section outlines the research design including the population, instrumentation, data collection, and issues of reliability and validity.

Participants

As purposeful samples should judge the likelihood of the research (Gall, Gall, and Borg, 2003), in spring 2012, the 14 SHSU graduate students who took the online Educational Multimedia course as part of their Master of Education in Instructional Technology program were invited to participate in this research. The cohort was made up of ten educators and four administrators who were part of the K – 16 educational systems. They were considered well grouped to shed light on the phenomenon of this study (Creswell, 2012) as they were educational technology specialist candidates who were in the educational systems. The participants' cooperation was sought to avoid coercion (Bogdan & Biklen, 2003).

Instrumentation

The participants were required to learn two free Web 2.0 game-programming applications including Scratch and Alice. They were also required to read one article on Project Based Learning (PBL) design for reflection and discussion. A three-week time frame was designated for learning Scratch followed by a two-week time frame for learning Alice. The instructional method was based on PBL. Each participant selected his/her own interesting subject to focus on the specific age of the targeted young students. An evaluation was conducted after each application was learned. Data were collected for each application by using an online discussion forum and a peer review process. Creswell (1998) stated that the natural setting for a qualitative inquiry is where the researcher gathers words or pictures to analyze them inductively in order to describe a process that is expressive and persuasive in language. A Facebook Group account was created to allow the participants to provide participant input. Each participant reviewed other participant's input and then provided his/her review.

The evaluation questions were posted as follows:

1. Discuss your learning experiences by using Scratch to create a game.
2. Discuss your learning experiences by using Alice to create a game.
3. Provide your reflection after reading the following article: Lynn, M. (2011). Reviving Interest in Math and Science. Duke Today News. Retrieved from http://today.duke.edu/2011/06/mathscience.

As qualitative design provided openness for discussions and findings (Bogdan and Biklen, 2003), these broad research questions provided a free space for the participants to share their personal experiences and invited the 14 participants to provide their peer reviews.

The assigned reading material shared the PBL design of promoting Alice in the middle schools. This article included a brief description of the learning gaps in math and science in the U.S. and supported the fact that Alice would create a fun learning environment by implementing game design into the school curriculum.

Data Collection

Through the semester, the personal email messages and Facebook messages were used as part of data collection. Once the qualitative data were collected from the Facebook, they were organized to form categories based on the terms used in the responses. By using a color coding

ISSN: 2167-1435
v29 n1937

system, the presence of each term was counted and analyzed by means of Frequency Analysis.

For Research Question 1 and 2, the following categories were found for each program: (a) entry behaviors, (b) main concepts, and (c) exit behaviors. These categories included terms like "frustrating," "difficult," "confusing," "anger" and "enraged", "excited," "challenging," "enjoyed," "interesting," "logic of the programming," "fun,"…etc. Under each category, the terms used frequently were tallied.

As a result, the findings were presented in both textual and numerical values (Onwuegbuzie &Teddlie, 2003).

Issues of Reliability and Validity

Reliability and validity are essential to the effectiveness of data collecting procedure (Best and Khan, 1998). Due to the nature of the online discussion format, there was no risk in facing methodological errors in the transcribed and translated data procedures compared to a face-to-face interview, and/or a focus group interviews (Onwuegbuzie & Daniel, 2005). The transcripts used in the study were originally written by the participants. When the messages were not clear, personal email messages and/or Facebook feedback postings were added to clarify the true intended messages resulting in trustworthy data collection and analysis.

Part of the data collection included participant feedback for the other members' peer reviews. Each member's viewpoints, feelings, and experiences were added correctly to the depth to the findings of this study measuring what the authors wanted to measure.

4. FINDINGS

This section will describe the findings of Scratch, Alice, and the impacts of PBL.

Before and during the programming sections, the participants shared their fears and worries about learning the programming software. Although the two participants had bachelor degrees in technology and minor in computer science, they expressed their worries for their outdated skills in programming.

Scratch

Nine out of 14 participants stated that Scratch was their first learning experience in programming to create a game. The main concepts the participants learned from the Scratch project were: basics of programming, inserting graphics, transitioning, variables and broadcasts, commands, brainstorming innovative ways, and troubleshooting.

When asked to focus on the entry behaviors at the beginning of learning Scratch, seven participants shared the feelings of "incredibly frustrating," and seven participants shared the experiences of "difficult and confusing." However, after the participants completed their Scratch project to create an educational game, the exit behaviors of an overall view for Scratch was overwhelmingly towards to a positive attitude and had a great challenging impact. The comment "I enjoyed learning the software" was shared for ten times. The comment "It was a challenging, yet interesting assignment" was shared for five times. One participant expressed: "I understand now about the logic of the programming!" Regarding the theme of educational impact, some members shared the following statements: "Creating a game was fun and educational...." "I think students would enjoy creating games using this software…" "This structured, methodical process of building a product that is fun, entertaining and helps learning…"

Alice

The main concepts the participants learned from Alice were 3D animation, camera functions, and logic functions. The participants expressed their feelings before they started to work on Alice in the following words and frequencies: (a) anger, and enraged (2 counts), (b) frustrating (1 count), (c) excited (3 counts), and (d) difficult (1 count). After the participants completed their Alice project, only one participant commented that it was a "fun" project. But ten participants commented that it was a "challenging assignment." Overall, they shared a view described as "good learning and rewarding experience." One participant, who had a hard time starting Alice, later wrote: "I love programming!" at the end of the project.

Regarding the theme of educational impact, some members shared their input as follows: "Igniting a student's curiosity is what it's all about!" "I think it was a great learning experience and would like to be able to use it in my classes on regular basis to help reinforce lessons taught previously to the students…." "I would definitely create some generic games that I could use and just change the questions…" "I can invasion myself using the program in the future…" "I think students would enjoy being able to create fun 3D games…" "I saw a very

ISSN: 2167-1435
v29 n1937

creative interpretation of The Outsiders, a novel read and dissected by many jr high students in Language Arts classes…."

Project-Based Learning Impact

In five weeks, the participants completed two game-designs by using Scratch and Alice. They then reviewed the article published by Duke Today News, entitled "Reviving Interest in Math and Science". Two themes merged from the participants' reflection and peer reviews: Transferable and Reflection Concepts.

Transferable Concept. Approximately 57 percent of input directly reflected the transferable concept of applying game-programming to various fields. Not only working through science or math problems with the game-programming software is transferable, but also it benefits the other subjects, such as language, art, social study, and other subjects. One participant stated that students might find in Alice a"fun" way to express their ideas and have their ideas and design presented in writing. One participant stated: "Being exposed to a task never before considered doable and working through a program such as Alice can truly enlighten the participants to ideas they never considered and abilities they would otherwise never approach."

Nine participants agreed: "Introducing Alice during the middle school years is the greatest way to engage young people into the world of programming." These findings lead us to believe that the game-programming might help stimulate the students' curiosity for technology. It is also believed that the fun part of game-programming software would help the youngsters to be more engaged in understanding a higher level language such as C++ or Java. Furthermore, the game-programming can go a long way in making subjects relevant to the students and give the students the ability to master what they learned.

In addition, for teachers, game programming software provides the additional benefits of allowing improved learning across subject areas. Lana Dyck (2009) stated that the outcomes showed a great learning curve by applying computers and technology into the curriculum from which the teachers created lesson plans by integrating programming into various subject areas including math, science, language arts as well as technology/programming. It was evident that creating and animating stories in Alice was a powerful tool in motivating students to learn (Lana Dyck, 2009).

A final thought from one participant included: "Students with the potential to be good at it should be introduced to it at an early age, so that their apprehension can be assuaged."

Reflection Concept. Approximately 43 percent of the reflections stated that they changed their thoughts about how to teach after having completed this project-based learning procedure. Twelve participants recognized a positive attitude when hands-on projects were applied in their classroom teaching practices. As one participant summed it up, from the outcome, it was evident that with PBL: "The students were more engaging and excited about learning and completing the project."

A 25 percent of the participants agreed that training teachers on how to use programs like Alice would impact students on the way they were taught. Applying PBL could provide the teachers an opportunity to be creative and to find alternatives to teach and to assess student understanding more effectively (Edutopia, 2009). To summarize this section, one participant commented:"by encouraging children to create animations or games that help others learn, they are demonstrating mastery of concepts themselves."

5. CONCLUSIONS

Resnick (2007) indicated that today's students "must learn to think creatively, plan systematically, analyze critically, work collaboratively, communicate clearly, design iteratively, and learn continuously" (p.6). The findings of this study indicated that using programming tools such as Alice and Scratch help the teachers and students to rethink about computer science which can be fun and challenging. Young-Jin (2011) stated that using a drag-and-drop method similar to Legos helps users to more likely become motivated as they look deeper into the possibility of linking the games with the teaching subjects. Gans and Lee (2012) confirmed that "multimedia stories can be crafted in language arts classes, multilevel computer games can be designed in mathematics, and art and music projects can be delivered through electronic media" (p.5).

Games are as applicable to young learners as they are to older game developers and useful tools for learning problem solving and higher level thinking. A new prospect for the game-

ISSN: 2167-1435
v29 n1937

programming application concerning teaching would be to implement the programming concepts through any subject areas. In the classroom with Scratch and Alice, or other programming languages, students can learn to bring together media as graphics and audio, and incorporate them into a program which tells a story or describes and extends a lesson through a narration or a game. Critical thinking skills are enhanced as they extend lessons.

This study found that Scratch and Alice can be categorized in the fine line of amateur programming languages which could fit into regular curriculum. Younger kids can do powerful things with Scratch and Alice once they have a better grip on the terms and the language. One of the participants stated: "it's guaranteed to be fun!" Hereafter, the research question can be answered as: "yes, the game-programming can bridge the students' learning gaps in all subjects!"

Moreover, while implementing various strategies to motivate U.S. students' interests in math and science, the government might need to quickly lounge the much needed and workable instruments to improve students' learning in STEM which would lead to also solving the problem concerning the much needed U.S. STEM workers. As U.S. STEM workers need to develop new skills and opportunities to distinguish themselves (Hira, 2009), it is recommended that we need to teach the students more analytically in order for their "technology skills to be retained and be valuable" (Lewis, 2009, p.3).

6. REFERENCES

Alice. (2012). Retrieved June 14, 2012, from Alice: http://www.alice.org/index.php?page =what_is_alice/what_is_alice

Baldwin, R. (2007). Learn to Program using Alice. RetrievedJune 14, 2012 from http://www.dickbaldwin.com/alice/Alice0100 .htm

Best, J.W., & Kahn, J.V. (1998). *Research in education*(8th ed.). Boston: Allyn and Bacon.

Betcher, C. (2010). Teaching kids to think using Scratch. ChrisBetchablog home page, Retrieved June 14, 2012, from http://chrisbetcher.com/2010/10/teaching-kids-to-think-using-scratch/

Bogdan, R.C., &Biklen, S.K. (2003). *Qualitative research in education: An introduction to theory and methods* (4th ed.). Boston: Allyn and Bacon.

Brown, A.S. & Brown, L.L. (2009). What are science & math test scores really telling U.S.? *The Bent of TAU BETA PI*. Retrieved June 14, 2012, from http://www.tbp.org/pages/ publications/Bent/Features/W07Brown.pdf

Buck Institute for Education. (2009). Does PBL work?. Retrieved June 14, 2012, from http://www.bie.org/research/study/does_pbl _work

Carnegie Mellon University. (2012). School of Computer Science. Retrieved September 18, 2012 from http://www.cs.cmu.edu/

Carter, L. (2006). Why Students with an apparent aptitude for computer science don't choose to major in computer science. *ACM*. New York. Retrieved June 14, 2012, from: http://www.imageofcomputing.com/pdf/p27 -carter.pdf

Cottrell, S. (2011). *Critical thinking skills: Developing effective analysis and argument*. (2nd Ed.). Basingstoke, Hampshire: New York.

Creswell, J.W. (1988). *Qualitative inquiry and research design: Choosing among five traditions.* Thousand Oaks, CA: Sage Publications.

Creswell, J.W. (2012). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research* (4th Ed.). Boston: Pearson.

Dann, W., & Cooper, S. (2009). Education Alice 3: Concrete to Abstract. *Communications Of The ACM*, 52(8): 27-29.

Education Week. (2012). U.S. Education pressured by international comparisons. Education Week home page. Retrieved June 14, 2012, from http://www.edweek.org/ew/ articles/2012/01/12/16overview.h31.html

Edutopia. (2009). Why Integrate Technology into the Curriculum: The Reasons Are Many. Retrieved June 13, 2012, from http://www.edutopia.org/technology-integration-introduction

Edutopia. (2012). PBL research summary: Studies validate project-based learning. Retrieved June 13, 2012, from

ISSN: 2167-1435
v29 n1937

http://www.edutopia.org/project-based-learning-research

Feldman, D.A. (2009). *Critical thinking: Make strategic decisions with confidence (*2nd ed.). USA: Axzo Press.

Gall, M.D., Gall, J.P., & Borg, W.R. (2003). *Educational research: An introduction* (7th ed.). Boston: Allyn and Bacon.

Gans, P. & Lee. I. (2012). K-8 CS education: Building a solid foundation. *CSTA Voice*. 8(4):4-5.

Giansante. (2009). Alice 3D programming. Lincoln Technology. Retrieved from http://www.lincoln.edu.ar/comp/alice/Alice.pdf

Hira, R. (2009). U.S. Workers in a Global Job Market. *Issues in Science and Technology*. National Academy of Sciences Audience. Retrieved June 14, 2012, from http://www.issues.org/25.3/hira.html

IDLBI, A. (2009). Taking Kids into Programming (Contests) with Scratch. Retrieved from Vilnius University Institute of Mathematics and Informatics: http://www.mii.lt/olympiads _in_informatics/pdf/INFOL040.pdf

Kelleher, C. & Pausch, R. (2007) Using storytelling to motivate programming. Communications of the ACM, 50(7), 58-64.

Koray, O. & Koksal, M.S. (2009). The effect of creative and critical thinking based laboratory applications on creative and logical thinking abilities of prospective teachers. *Asia-Pacific Forum on Science Learning and Teaching*. 10(*1*), 1-14.

Lamb, A., & Johnson, L. (2011). Scratch: Computer programming for 21st Century learners. *Teacher Librarian*, 38(4), 64-68.

Lana Dyck, S. R. (2009). Connecting Computer Science and the K to 12 Classroom. Retrieved from The Computing Research Association: http://www.cra.org/Activities/craw_ arhive/dmp/awards/2009/Dyck/files/final Report.pdf

Lawson, A.E., Banks, D.L. Ve Logvin, M. (2007). Self-efficacy, reasoning ability and achievement in college Biology. *Journal of Research in Science Teaching*. 44(*5*), 706-724.

Lewis, D. (2009). Attracting the next generation of students to computing. *CSTA Voice*. 5(*2*),1-3.

Lynn, M. (2011). Reviving Interest in Math and Science. *Duke Today News*. RetrievedJune13, 2012, from http://today.duke.edu/2011/06/mathsicence

MIT. (2012, January 1). Scratch. Retrieved June 14, 2012, from About Scratch: http://info.scratch.mit.edu/About_Scratch

Oddie, A., Hazlewood, P., Blakeway, S., & Whitfield, A. (2010). Introductory problem solving and programming: robotics versus traditional approaches. *Innovations In Teaching & Learning In Information & Computer Sciences*, 9(2), 86-96.

Onwuegbuzie, A. J., & Daniel, L. G. (2005). Editorial: Evidence-based guidelines for publishing articles in Research in the Schools and beyond. Research in the schools, 12(2), 1-11.

Onwuegbuzie, A.J.,& Leech, N.L. (2005, March 10). A Typology of Errors and Myths Perpetuated in Educational Research Textbooks Current Issues in Education [On-line], 8(7). Retrieved June 13, 2012, from http://cie.ed.asu.edu/volume8/number7/

Onwuegbuzie, A.J.,& Teddlie, C. (2003). A framework for analyzing data in mixed methods research. In A. Tashakkori& C. Teddlie (Eds.), *Handbook of mixed methods in social & behavioral research* (pp.351-382). Thousand Oaks, CA: Sage Publications.

PISA. (2009). Strong Performers and Reformers in Education: Lessons from PISA for the United States. Retrieved August 20, 2012, from http://www.oecd.org/pisa/46623978. pdf

Project Lead the Way. (2012). Gateway To Technology: Middle School Engineering Program. RetrievedJune 14, 2012, from http://www.pltw.org/our-programs/middle-school-engineering-program

Project Lead the Way. (2012). Outcomes. Retrieved June 20, 2012, from http://www.pltw.org/educators-administrators/outcomes

Ravitz, J., Mergendoller, J., Markham, T. (2004). Online professional development for project based learning: Pathways to systematic improvement. *Association for Educational Communications and Technology*. RetrievedJune 14, 2012 from Alice: http://www.freewebs.com/siowyy/Online%20PBL.pdf

Redmond, A., Thomas, J., High, K., Scott, M., Jordan, P., & Dockers, J. (2011). Enriching Science and Math Through Engineering. *School Science & Mathematics*, 111(8), 399-408.

Resnick, M. (2007, December). Sowing the Seeds for a More Creative Society. *Learning & Leading with Technology.* Retrieved June 14, 2012, from http://web.media.mit.edu/~mres/papers/Learning-Leading.pdf

Resource Area for Teaching (RAFT). (2009). A Case for Hands-on Learning. Retrieved June 30, 2012 from http://www.raft.net/case-for-hands-on-learning

Science 2011: National Assessment of Educational progress at Grade 8. *National Center for Education Statistics*. Retrieved June 14, 2012, from http://nces.ed.gov/nationsreport card/pdf/main2011/2012465.pdf

Scratch. (2012). Retrieved June 14, 2012, from http://info.scratch.mit.edu/About_Scratch

Sezen, N. & Bulbul A. (2011). A scale on logical thinking abilities. *Procedia Social and Behavioral Sciences 15.* (2011) 2476-2480.

Smiha, R. (2012). Computer Science Careers. *CSTA Voice*. 7(*6*), 1-2.

Sykes, E. R. (2007). Determining the effectiveness of the 3D Alice programming environment at the computer science I level. *Journal Of Educational Computing Research*, 36(2), 223-244.

Tashakkori A., &Teddlie, A. (Eds.). (2003). *Handbook of mixed methods in social & behavioral research.* Thousand Oaks, CA: Sage Publications.

Tangney, B., E. Oldham, C. Conneely, E. Barrett, & J. Lawlor (2010). Pedagogy and processes for a computer engineering outreach workshop – the B2C model. IEEE Transactions in Education. 53(*1*), 53-60.

Texas Education Agency. Chapter 74. Curriculum requirements subchapter D. Graduation requirements, beginning with school year 2001 – 2002. Retrieved September 18, 2012, from http://ritter.tea.state.tx.us/rules/tac/chapter074/ch074d.html

Texas Education Agency. Chapter 74. Curriculum requirements subchapter E. Graduation requirements, beginning with school year 2004 – 2005. Retrieved September 18, 2012, from http://ritter.tea.state.tx.us/rules/tac/chapter074/ch074e.html

Texas Education Agency. Chapter 74. Curriculum requirements subchapter F. Graduation requirements, beginning with school year 2007 – 2008. Retrieved September 18, 2012 from http://ritter.tea.state.tx.us/rules/tac/chapter074/ch074f.html

Texas Education Agency. Chapter 74. Curriculum requirements subchapter F. Graduation requirements, beginning with school year 2012 – 2013. Retrieved September 18, 2012 from http://ritter.tea.state.tx.us/rules/tac/chapter074/ch074g.html

Texas Higher Education Coordinating Board. (2012). Home page. Retrieved June 14, 2012 from http://www.thecb.state.tx.us/index.cfm?objectid=6AB82E4B-C31F-E344-C78E3688524B44FB

Utting, I., Cooper, S., Kolling, M., Maloney, J., & Resnick, M. (2010). Alice, greenfoot and scratch. *Transactions on Computing Education*. 10(4).

Villaverde, K., Jeffery, C., Pivkina I., (2009) Cheshire: Towards an Alice based game development tool", Proceedings of the International Conference on Computer Games, *Multimedia & Allied Technology*, p. 321-328. Retrieved from EBSCO host.

Young-Jin, L. (2011). Scratch: Multimedia programming environment for young gifted learners. *Gifted Child Today*, 34(2), 26-31.