

Weaving Computing into all Middle School Disciplines *

Susan H. Rodger
Duke University
Durham, NC USA
rodger@cs.duke.edu

Dwayne Brown
Duke University
Durham, NC USA
dwayne.brown@duke.edu

Michael Hoyle
Duke University
Durham, NC USA
michael.a.hoyle@gmail.com

Daniel MacDonald
Duke University
Durham, NC USA
dkm12@duke.edu

Michael Marion
Duke University
Durham, NC USA
michael.marion@duke.edu

Elizabeth Onstwedder
Duke University
Durham, NC USA
eho@duke.edu

Bella Onwumbiko
Duke University
Durham, NC USA
beo3@duke.edu

Edwin Ward
Duke University
Durham, NC USA
emw35@duke.edu

ABSTRACT

In order to get students interested in computing, we teach middle school teachers of different disciplines programming with Alice and work with them on integrating computing into their discipline. Alice provides an interface for novices to create animations easily and quickly, which is attractive to and fun for students. We have been developing Alice curriculum materials for integrating computing into middle school disciplines for six years. Although our target audience is middle school, our materials are used by teachers from elementary school to introductory college level. This paper describes our newest curriculum materials for several disciplines developed by both us and our teachers. Our newest curriculum materials include tutorials, sample projects, and *challenges*, which are projects with missing pieces. We also discuss our recent outreach efforts with middle school students.

Categories and Subject Descriptors

K.3.2 [Computing Milieux]: Computers and Education-Computer and Information Science Education

General Terms

Human Factors

*The work of these authors are supported by the National Science Foundation grant DRL-1031351 and IBM Faculty Awards

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ITiCSE'14, June 21–25, 2014, Uppsala, Sweden.
Copyright 2014 ACM 978-1-4503-2833-3/14/06 ...\$15.00.
<http://dx.doi.org/10.1145/2591708.2591754>.

Keywords

Alice, virtual worlds, K-12 education, middle school, introductory computer science

1. INTRODUCTION

Although middle school students use computers constantly throughout the day, the majority of them do not understand the meaning of computer science. Their exposure is likely not computational thinking or algorithmic problem solving, but instead using software such as word processors, presentation tools, and online course environments such as Moodle. A report in 2010 [1] produced by the Association of Computing Machinery (ACM) and the Computer Science Teachers Association (CSTA) stated that computer science education in the United States is at a crisis. While the digital age is transforming the world, U.S. K-12 education is not preparing students with the fundamental computer science knowledge to be prepared to take on jobs in the more computing-intense future. The NCWIT Scorecard [15] mentions that the U.S. Bureau of Labor Statistics predicts a total of 1.4 million computing-related jobs by 2018 and that one-third of them will be unfilled because not enough students are choosing computer science as a major.

There are several difficulties in moving toward teaching computer science in middle schools. One such difficulty is the lack of computer science standards. The CSTA and ACM have developed K-12 computer science standards (revised in 2011) [20] and a number of other documents that link their standards with other national standards. Another difficulty is teacher certification. Recently the CSTA has developed a document [13] about the lack of teacher certification for computer science in the U.S. showing a state by state report and many flaws in the existing teacher certification and licensure processes.

There are many efforts to increase the number of K-12 students exposed to computing such as the CS10K project [8], the new AP CS Principles Course [6], and a new effort with code.org [5] that has gotten millions of students trying an hour of programming by using their one-hour tu-

torials. Many projects target middle school including CS Unplugged [7], Alice [2] and Scratch [19], and some projects are focused on a specific discipline such as integrating mathematics with computing at the K-12 level [4, 12].

This paper describes our efforts on integrating computing with all disciplines using the Alice programming language. There are many books on Alice programming including one by Dann and others [9], and papers describing Alice curriculum materials such as one on Alice projects for middle school disciplines [17]. There have been three Alice Symposiums with the most recent one in 2013 [3] that show many ways how people are using Alice from elementary school to college level. For example the paper [21] shows how students at a camp create Alice videos and then use green screens to splice in videos of the students into their Alice videos. There are studies that show Alice aids in learning computer science such as [14] that shows college students using Alice had improved performance in CS 1, retention and attitudes about computer science. In [22], they show that students in a middle school game-programming course demonstrated an understanding of a range of computer science concepts while using Alice to make games.

In this paper we describe the curriculum materials we have developed for learning Alice programming, and how we work with teachers on integrating computing with Alice into all middle school disciplines. Although there are a few middle schools that have programming courses [22, 23, 10], most middle schools in the United States do not have a programming course or room to create one, and most teachers have no programming experience. Efforts such as code.org are reaching millions of students for one-hour, but the teachers are not being trained to support those efforts. In our work we run professional development to teach middle school teachers in all disciplines with no knowledge of programming how to program with Alice and help them prepare lesson plans that integrate computing with their discipline. Our curriculum materials include tutorials, sample projects, and *challenges*, which are projects with missing pieces. We also discuss our recent outreach efforts with middle school students.

2. BACKGROUND AND WORKSHOPS

We have been teaching Alice at the college level at Duke University for over twelve years. About seven years ago we started teaching Alice to middle school students. After seeing their excitement with Alice we started teaching Alice to K-12 teachers and started developing curriculum materials for those teachers. We hold two-week workshops in the summer to teach the teachers programming with Alice and to work with them on developing lesson plans. The teachers are mostly middle school, with some also at the elementary level and the high school level (but not programming teachers). Our focus is on teaching beginners in different disciplines. Since 2008, we have taught Alice to over 200 K-12 teachers. Those teachers are from the disciplines language arts, history, science, math, english as a second language (ESL), foreign language, music, art, business, media and physical education. Our curriculum materials and the lesson plans the teachers develop at our workshop are all available for free at our website [11]. We have over 130 tutorials and example worlds and over 170 lesson plans developed by teachers attending our workshops that are available for free. Although our target is middle school, the materials are being used by

teachers at all levels in K-12 and even some colleges. We use many of the materials at Duke University in a semester Alice course CompSci 94. According to Google Analytics, since September 2012 when we started tracking visitors we have had over 26,000 visits to our Alice materials website of which over 19,000 were unique visitors.

3. CURRICULUM MATERIALS

We have developed five types of Alice curriculum materials for students to use in learning Alice, and for teachers to use in supplementing their lesson plans. We have created beginner tutorials, topical tutorials, enhanced objects, challenges, and projects. We have created two documents available on our web site to show how our Alice materials map with the CSTA standards mentioned earlier and for our math materials with the Common Core Mathematics Standards [16] in the United States. Forty-five states have adopted the Common Core State Standards. Many of our tutorials list the standard on the tutorial.

3.1 Getting Started Tutorials

Our simplest tutorials are beginner or *getting started* tutorials. These tutorials are essential for a teacher in a different discipline to be able to teach basic programming concepts in a short period of time to students. We have nine beginner tutorials to choose from that range from short 15 minute tutorials to introduce Alice, to multipart four-hour tutorials that put together many concepts. Based on feedback from teachers attending our workshop and our own experience of teaching the teachers these tutorials, we recently created simpler versions of many of our beginner tutorials. For example, we converted a four-part tutorial with a princess and dragon theme into a shorter three-part tutorial with simpler parts. It is exciting to teach a dragon how to flap its wings elegantly if one has a lot of time, but using a simpler quick up and down motion on the main part of the wing gets the point across in a lot less time. Since our teachers are mainly discipline-specific teachers they need shorter tutorials. As another example, we created a shorter version of one of our one-hour tutorials by removing the multiple camera views, and created a separate tutorial on moving the camera that could be added on at a later time.

The majority of our tutorials are in .pdf format. Based on teacher feedback we created a new one-hour getting started tutorial that is in video format, divided into six videos. The tutorial is about a person on an island who uses a rowboat to escape a shark. Teachers said there are times when some of their students have 10 or 15 minutes free and they needed shorter videos. We also provide finished worlds of each part that can be used to start a new part if needed. We have started converting our other tutorials into videos.

3.2 Topical Tutorials

We created *topical tutorials* on either computer science topics or animation topics. Our computer science topics include tutorials on typical programming topics such as conditionals, loops, methods, functions, parameters, inheritance, variables, randomness, loops, and events. However we use different names for the tutorials such as "making decisions" for the tutorial on conditionals and "creating a fancier chicken" for the tutorial on inheritance. We have recently created new tutorials on lists and arrays.

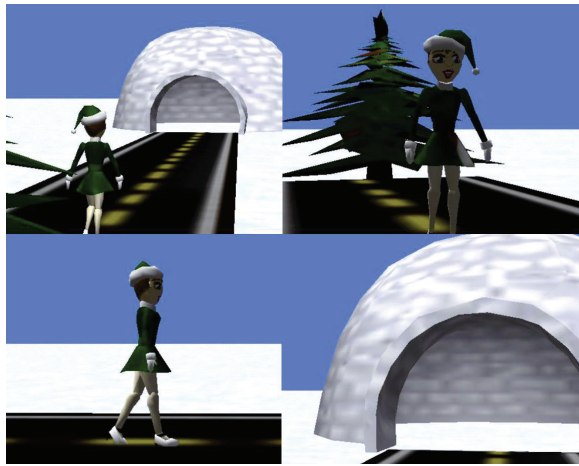


Figure 1: Showing four camera views

Our other topical tutorials are animation concepts. These cover topics such as moving the camera, changing scenes, adding in pictures, changing the texture of objects, and changing the lighting. With feedback from our teachers we modified previous tutorials. For example, we have a scene change tutorial that shows you how to setup an animation with three scenes, one in the dessert, one in the water and one on the moon, and how to fade to black and then fade in to another scene. We modified this tutorial to add a character and showed the teachers how to setup the location and orientation of the character in each scene, and then how to move the character between those set positions. We also created new animation tutorials requested by teachers. One of our new tutorials is on setting up multiple camera views that follow a particular person as they move through a world. Figure 1 shows four of the camera views that follow the girl as she walks along the road towards the igloo. First the camera is behind the girl, then looking from the side of the girl, then looking at the girl, and then is the girl's view. You can see the difference in her distance from the igloo as she walks towards it.

3.3 Helper Objects

We created *helper objects* to help teachers and students create an animation faster, instead of creating everything from scratch. The helper object is a 3D object that may be invisible and already has several functions and methods. We provide short tutorials that show how to use the functions and methods that come with the object and suggestions for possible modifications. A student can import the object into their world and start using the functions and methods. The first helper objects we created were for teachers to help them create quizzes. After importing a quiz object, the teacher can call a method with a question and answer passed in as arguments to parameters.

Based on requests from teachers we have created several new helper objects. We have a timer object that already comes with a start, pause and initialize method and displays the current number. The timer counts down from the starting value. We have a scorekeeper object that allows one to set a score, increment the score (by any amount) and decrement the score. This object could be used in a game or quiz. We have a fader object that fades out and fades into



Figure 2: Start of Wizard Challenge

scenes similar to our scene tutorial mentioned earlier, but this object comes with many of the methods already built.

3.4 Challenges

A challenge is an Alice world that is mostly built but contains missing pieces. Each challenge comes with a partially built Alice world and a handout describing the missing piece, the challenge! We expect students to know some basic programming before they do the challenge, such as our beginner tutorials, and some challenges may require additional topical tutorials. Most of the challenges are short so they can be completed as a short exercise in a class, another request from teachers. A challenge allows a student to focus on a particular topic and after implementing just that part have an interesting animation. Building the complete animation may take too long for them to do in class.

3.4.1 Wizard World Challenge

Our Wizard World challenge is our largest challenge world and is actually eight challenges in one world. It is a combination of math and programming challenges and the theme uses courses one might take in the school Hogwarts in the Harry Potter series by J.K. Rowling [18]. Our world is titled "Hailey Programmer and the Goblet of Java." The start of the animation shown in Figure 2 shows the eight levels for this game.

Here is how a general level works. When a level is clicked on, a method for that level runs and calls the method the student is suppose to modify. It then checks to see if the animation tasks were completed. If they were not, the student sees the animation does not work correctly and is given instructions and the name of the method they need to edit. Only the methods the student needs to edit are displayed at the top level. The program was designed with the code the student does not need to edit hidden away in folders. Additional tutorials are suggested before attempting some of the levels. The additional tutorials can be found on our website.

We describe the first level in detail which is a coding challenge. We will refer to the student in the animation as the "student" and the student completing the challenge as the "MS student". The MS student starts the animation and



Figure 3: Charms Class

the only level that does not have a password is the charms class, so they select it. This level focuses on basic code for moving and turning objects and the start of it is shown in Figure 3. The student in the animation is trying to use her magic wand to make the three objects rise and to turn the lion. She tries and nothing happens. The MS student is instructed to write code in the method `world.charms` to move and turn the objects. The MS student then stops the animation and edits the method `world.charms`. This method has comments describing the tasks the MS student needs to complete. The MS student adds code and runs the Alice world again selecting the charms level. If the code is correct, the objects rise and turn and the MS student is rewarded with a password for the next level.

Level 2 is the History of Programming class and is also a coding challenge. In this level an instructor is talking at the front of the classroom while three students are watching and also playing pranks. The MS student needs to modify three sections of code in `world.history` to make an invisible object visible to reveal a message (changing a property), to make the teacher rise by the teacher's height (using a built-in function), and to make a chair bounce repeatedly (using a loop).

Level 3 is Flying class and is a math and programming challenge. The student must understand how a 2D coordinate system works. In the animation a picture of a coordinate plane is shown and the student has a broom. Their challenge is to fly to different coordinates on the grid by moving up, down, forward and backwards. For a sequence of coordinates, the MS student must add code to call functions for moving up, down, forward and/or backwards and pass the appropriate arguments for the number of units to move. For each new coordinate pair, the student with the broom must move from the current position to the new position. The first few coordinates are number locations such as to location (3,2), but then later coordinates use algebraic equations such as $(2, 2x+1)$.

Level 4 is Potions class and is a math and programming challenge. Here students will add ingredients to a potion using the correct ratios and calling functions with arguments. Level 5 is Defense Against the Dark Arts class and focuses on geometry and programming. Here one will need to defend themselves by calculating distances such as the distance in a semicircle around a troll, or along the edge of a right

triangle. Level 6 is Divination class and focuses on drawing five cards to read their fortune and to calculate the odds of getting a royal flush. This level uses a list of cards. They would need to complete the list tutorial before attempting this challenge. Level 7 is Herbology. Students will use their knowledge of lists and population sampling to calculate the difference in height between the average height of a group of trees and the average height of a group of flowers. Students will need to know the concept of random sampling for this level. Level 8 is the final level. Students combine all the programming knowledge they have learned to save the golden dragon's egg from the giant troll to win the Tri-programmer Tournament.

3.4.2 Calculator Challenge

The calculator challenge started with a request from a math teacher attending our workshop for a calculator object. We forwarded the request to the Alice team and they made a calculator object with button parts. We then wrote the code to allow for one to enter numbers into the calculator and to display those numbers, and wrote functions for all the button operators. The calculator challenge provides our version of the calculator with the functions missing code. Before programming, the students can run the world and enter numbers but will see that buttons such as add do not work. Students then add the code for add, subtract, multiply, divide, square root and log to get the calculator to work. They can then create their own buttons for additional operators, make the button by creating an image and gluing the button onto the calculator, and then add a function for this button. One of the teachers attending a followup workshop showed us a calculator one of her students made who had added almost twice as many buttons and corresponding functions to his calculator.

3.4.3 Boat Racing Game Challenge

The Boat Racing Game is a game in which a student uses arrow keys to drive a boat through ten arches before the time is up. We have taken this game and built on several challenges, all that collect and analyze data. In one example challenge the arches move to random locations before the game starts and code has been added to collect in a list the distance traveled between pairs of arches. The student needs to then add code to use the data to calculate the total distance in order to calculate the average distance between pairs of arches. In another challenge, the student plays the game multiple times, saving each time in a list, and then computes the average time to complete the game.

3.4.4 Pythagorean Challenge

One challenge is to compute distances between objects placed on a 2-d coordinate system. The task is to move a character from his current location to other objects by computing the distance from their current location to the object they want to move to. To finish this challenge the student must modify one function to compute the distance between two points and to use the pythagorean theorem to compute this distance.

3.4.5 Science Challenge

An example of a science challenge for Biology is an Alice world on Punnett squares. Scientist use Punnett squares to predict the outcome of breeding experiments. In this

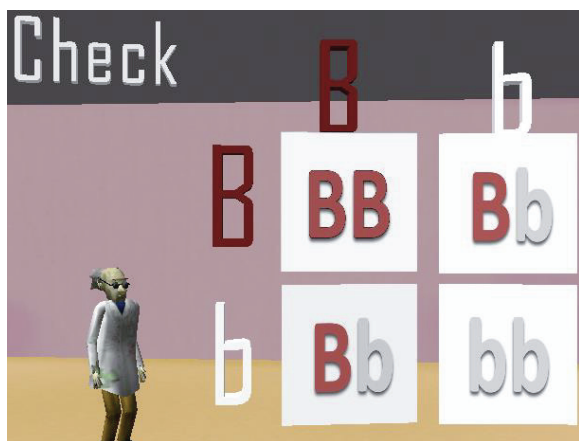


Figure 4: Calculating Punnett Squares

challenge the student has to create four events and complete a method to rotate between four images. When the completed world is run, the user is asked to compute the Punnett Squares for a problem, and they will click on four squares that each list four possible choices (images). For each square cycle through the four images until you find the correct answer as shown in Figure 4 in the top left square. Once correct, the student is then asked related probability questions.

3.5 Sample Projects

We have created over twenty sample project tutorials for a variety of disciplines that tell a story or play a game. The teachers wanted to see projects that put together in a cohesive story or game many of the topics they learned in the workshop. Projects are used heavily in middle school and are the most likely place for teachers to integrate computing. Instead of creating a poster, a power-point presentation or a model for a project, a student could create an Alice animation for a project. The animation could tell a story and also interactively ask quiz questions. Previous projects we created include a sample book report for a language arts class, a quiz on the sounds of different instruments for a music class, a cooking demo to make bread in spanish for a foreign language class. Here we describe some of the new project tutorials.

For science we describe two new projects. One is a tutorial for building an Asteroids game that is like the 1979 Atari classic video game. The user pilots a ship in space and uses the spacebar to shoot a laser at incoming asteroids. The goal is to hit all the astroids before one of them hits the ship. Another new science project is a tutorial that shows how to build an animation of the solar system. In this tutorial the student makes the planets by texturizing shapes and then programs the planets to move around the sun. The lighting is changed in the world to radiate from the sun.

A new project tutorial for a keyboarding class is a keyboard racing game with two characters on bikes who race to the finish line. One of the bikes only moves forward if letters that are displayed are correctly typed. The player is trying to beat the other bike which is moving steadily forward. This project comes with a starter world that already has all the objects you need in it.

3.6 Sample Worlds

We have created over 50 sample worlds to give students ideas on creating Alice worlds. The majority of the most recent sample worlds are for math topics.

Zombie soccer is an interactive world that gives math problems involving negative numbers. For a correct answer the soccer ball will score a goal and for an incorrect answer the zombie goalie will stop the ball. The instructions make it easy to alter the world to different levels of math. *Binary Game* is a game with penguins who have zeros or ones on their bellies. A decimal number is shown and the student has to click on the penguin to change the numbers on the penguin to compute the binary equivalent number. *Blaster* is an interactive game for shooting asteroids with numbers on them. In particular you want to only shoot the asteroids with the number that is the correct solution to a multiplication problem displayed. *Order of operations* is an interactive world in which an expression is displayed involving addition, subtraction, multiplication and division, and the student has to click on the operators in the order they should be calculated, and also has to give the result when applying each operation. After each operation the equation is simplified with the new result. *Fraction world* is an interactive world in which two fractions have an operator applied to them. The student is prompted to simplify the fractions first, and then to compute the answer, and finally simplify the answer if needed. *Bird graphing* allows a student to enter in a function in the code and then watch an animation of dots appearing on a graph to show the values of the function.

An example world for a keyboarding class is *Typing of the dead* game. Zombies with letters on their head come running towards you. They disappear if you type the letter displayed on their head before they get to you.

4. TEACHER LESSON PLANS

The teachers attending our workshops have created over 170 lesson plans that are available for use on our website. The lesson plans are categorized by discipline and within discipline by grade level. Each lesson plan has a title, the name of the teacher, the year it was created, a link to the lesson plan (usually in .pdf and .doc) and a link to the started and/or finished Alice world.

We give examples of two of the lesson plans from 2013. First, a history lesson for sixth graders on ancient Central America. The NC Standard is 6.TT.1, to use technology and other resources for the purpose of accessing, organizing, and sharing information. For this lesson the students will have already had one lesson with Alice and will first review some Alice and review the definition of civilizations. The students will then complete a short Alice tutorial to learn how to create billboards, sounds and 3D text in Alice. Students will view a short documentary and then research ancient civilizations in Central America. They will be given a sample Alice world to complete. They will fill in two scenes with billboards and information they obtained in their research.

Second we describe a lesson plan for a language arts class in 6th grade on fairy tales. Prior to class the students will read a collection of fairy tales and will have completed an Alice beginner tutorial. For this lesson, the students will analyze the fairy tales by identifying recurring themes. The teacher will show them an Alice world example of a fairy tale. The students will outline a story map for an original

fairy tale including standard elements and fairy tale themes. Then the students will build an Alice world for their story.

5. OUTREACH

Although our focus is on teaching teachers Alice, we describe some of our outreach activities with students. In November 2012 we were invited by a teacher who took our summer workshop to visit his Middle School and we taught Alice to three sixth grade math classes, about 75 sixth graders total. Each class was about an hour long. We taught them a tutorial with a character on an island who jumps into a boat and rows away. Then we provided them with some of our sample Math worlds to play including the *Order of Operations* and the *Fraction World*. The students were clearly excited about using Alice. The volume level in the room dramatically went up when they added an event so they could control the boat with the arrow keys. In those classes where we finished earlier and the students had more free time with Alice, they added objects to the world and expanded the animation. They asked us questions on how to do certain actions.

In March 2013, we held two three-hour activity days for sixth graders on Alice, and had a total of 26 students attend one of the sessions, with a balance of gender. We taught the students Alice using the same tutorial as before (these students were from different schools than the school we visited earlier). After some free time, they ran the *Order of Operations* and *Fraction World*. Then they worked on the calculator challenge we described earlier. In a survey given at the end of the sessions, the majority of the students thought Alice was easy to use and noted that Alice helped them learn about programming. More than half of the students thought Alice would be fun and valuable to use in their classes for projects and presentations.

6. CONCLUSION

We have developed several types of Alice curriculum materials for integrating computing into several disciplines. We provide professional development for teachers in the summer to teach them Alice and to work with them in creating lesson plans. Our training provides them with confidence and the lesson plans they develop are ready to use. Our teachers are now giving Alice presentations and posters at conferences. One of our teachers developed a full Alice course using our materials and won Teacher of the Year at her school. Our materials and the teachers' lesson plans are available for free.

7. REFERENCES

- [1] ACM and CSTA. Running on empty: The failure to teach k-12 computer science in the digital age, 2010. www.acm.org/Runningonempty/.
- [2] Alice. website, 2014. Retrieved January 1, 2014 from <http://alice.org>.
- [3] Alice Symposium June 2013. website, 2014. Retrieved January 1, 2014 from <http://www.cs.duke.edu/csed/aliceSymposium2013>.
- [4] Bootstrap. website, 2014. Retrieved January 1, 2014 from <http://www.bootstrapworld.org>.
- [5] Code.org. website, 2014. Retrieved January 1, 2014 from <http://code.org>.
- [6] College Board. CS principles, 2014. Retrieved January 1, 2014 from <http://csprinciples.org>.
- [7] Computer Science Unplugged. website, 2014. Retrieved January 1, 2014 from <http://csunplugged.org>.
- [8] J. Cuny. Finding 10,000 teachers. In *CSTA Voice*, 5, 6, pages 1–2, Jan 2010.
- [9] W. Dann, S. Cooper, and R. Pausch. *Learning to Program with Alice, Third Edition*. Pearson, Prentice Hall, 2011.
- [10] C. Distler. Piloting alice in the upper school. In *The Proceedings of The Alice Symposium 2013*, 2013.
- [11] Duke University Adventures in Alice Programming. website, 2014. Retrieved January 1, 2014 from <http://www.cs.duke.edu/csed/alice/aliceInSchools>.
- [12] E. Freudenthal, M. Roy, A. Ogrey, T. Magoc, and A. Siegel. Mpct: Media propelled computational thinking. In *The Forty-first SIGCSE Technical Symposium on Computer Science Education*, pages 37–41, 2010.
- [13] K. Lang, R. Galanos, J. Goode, D. Seehorn, and F. Trees. Bugs in the system: Computer science teacher certification in the u.s., 2013. CSTA Report.
- [14] B. Moskal, D. Lurie, and S. Cooper. Evaluating the effectiveness of a new instructional approach. In *Thirty-Fifth SIGCSE Technical Symposium on Computer Science Education*, pages 75–9, 2004.
- [15] National Center for Women & IT. Newit scorecard 2011: A report on the status of women in information technology, 2011.
- [16] C. o. C. S. S. O. National Governors Association Center for Best Practices. Common core state standards math, 2010. National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington D.C.
- [17] S. Rodger, M. Dalis, C. Gadwal, J. Hayes, P. Li, L. Liang, F. Wolfe, and W. Zhang. Integrating computing into middle school disciplines through projects. In *The Forty-third SIGCSE Technical Symposium on Computer Science Education*, pages 421–426, 2012.
- [18] J. K. Rowling. *Harry Potter and the Sorcerer's Stone*. Bloomsbury, 1997.
- [19] Scratch. website, 2014. Retrieved January 1, 2014 from <http://scratch.mit.edu>.
- [20] D. Seehorn, S. Carey, B. Fuschetto, I. Lee, D. Moix, D. O'Grady-Cunniff, B. Owens, C. Stephenson, and A. Verno. Csta k-12 computer science standards, revised 2011, 2011.
- [21] J. Shanahan and D. Marghitu. Software engineering java k12 outreach course with alice and cloud computing. In *Proceedings of the 2013 Alice Symposium*, 2013.
- [22] L. Werner, S. Campe, and J. Denner. Children learning computer science concepts via alice game-programming. In *The Forty-third SIGCSE Technical Symposium on Computer Science Education*, pages 427–432, 2012.
- [23] L. Werner, J. Denner, et al. Pair programming for middle school students: Does friendship influence academic outcomes? In *The Forty-fourth SIGCSE Technical Symposium on Computer Science Education*, pages 421–426, 2013.