CTI Charlotte Teachers Institute
Collaborative Teacher Education

Computer Programming is a SNAP! to Learn for Young Minority Females

By: Richard Whitehead, 2014 CTI Fellow
Northwest School of Arts

This curriculum unit is recommended for:
AP Computer Science, MaWD and CTE – Business Education
Grades 6-12

Keywords: Snap!, Computer Science, Programming, Minority Females, Women of Color, and Computer Programming

Teaching Standards: See Appendix 1 for teaching standards addressed in this unit. (Insert a hyperlink to Appendix 1 where you've stated your unit's main standards. For directions on how to insert a hyperlink, see Fellows Handbook, p. 29.)

Synopsis: This curriculum unit will introduce students to the world of computer programming through the visual programming language called Snap!. Students will discover and explore how computer programming allows them to create games and visual art through graphics along with using this visual language to communicate their ideas in a way that the computer can understand their commands in a fun and exciting way. The unit attempts to focus on minority females because of the lack of diversity in Computer Science (CS) today and the downward trend of representation by women of color over the last decade. Although this industry is the fastest growing sector for jobs in the United States, the stereotypes about computing as a discipline has turned women away from this industry over the last several decades. This has not always been the case for women. We, as educators, must address and reverse this negative perception of CS in America, otherwise very few young, minority women will have the aptitude or desire to benefit from the fruits of this expanding industry sector.

I plan to teach this unit during the coming year in all my CTE courses to 122 students in Multimedia and Webpage Design and CTE – Business Education courses for 6-12 graders.

I give permission for the Institute to publish my curriculum unit and synopsis in print and online. I understand that I will be credited as the author of my work.

Computer Programming is a SNAP! to Learn for Young, Minority Females

Richard Whitehead

Rationale

This curriculum unit is called *Computer Programming is a SNAP! To Learn for Young, Minority Females*. Snap! can be used for all students, but my focus will be minority females for reasons I will explain throughout this curriculum unit. SNAP! is a visual computer programming software, pioneered by the Computer Science Division of the College of Engineering at the University of California-Berkeley, which we utilize for the introduction of computer programming to all students, grades 6-12, at Northwest School of the Arts (NWSA) in Charlotte. NWSA is the only secondary magnet school in the Charlotte Mecklenburg School (CMS) system to offer a concentration in the Arts. NWSA attracts artistic-minded students from the entire CMS system. All students are required to pass an audition, for their applied art specialty, during the spring and/or summer proceeding the fall semester of the following school year before they can be accepted into any program of study at NWSA. NWSA offers concentrations in Music (Instrumental and Vocal), Piano, Dance, Theater, Costume Design and the Visual Arts. NWSA has an extremely diverse student population. NWSA consists of over 65% female students of which approximately 75% of those females are minorities. Despite a high minority population, NWSA has the highest graduation rate in CMS. Several factors contribute to that success. It begins with an enthusiastic student population who truly want to be at NWSA, therefore making NWSA students more engaged than the “normal” public school population.

NWSA students are eager to learn, have extremely creative minds and exhibit great imagination. Naturally, these positive characteristic traits can help most NWSA students with the comprehension of abstract ideas. Since computer programming is based on algorithms, NWSA students show a certain aptitude for creating well balanced and very logical programming and design ideas. The most difficult hurdle in teaching this curriculum is overcoming the typical stereotypes associated with Computer Science (CS) and what computing equates to in the 21st century. As we all understand, perception can be reality for most students and, unfortunately, there tends to be little difference about CS at NWSA with our largely minority female student population than at any other school around our system, in N.C. or even around our entire nation. The reality is that the trend over the past 15 years or more has been a sharp and pronounced decline in females and minority females in CS in the United States.

Computer Science Today

This summer I attended 3 computer programming workshops at N.C. State University and the University of Virginia focused exclusively on attracting more female and minority female students into computing and CS in some academic capacity. I found that, in the U.S., the national average for females employed in any type of CS career in 2011 was at about 25% of which only 3% were African-American women, 4% were Asian women and only a meager 1% of the computing workforce included Hispanic women.[1] It is estimated that there will approximately 1.4 million jobs in CS in the United States by the year 2020 caused by retirement of existing employees and the need for an influx of new employees in the industry. Of these 1.4 million job openings it is also estimated that the U.S. computing graduates will only be able to fill an estimated 30 percent of these openings[2]. These technology jobs are some of the highest paying jobs in the world. If U.S. companies can't fill them with Americans they will be forced to seek foreign employees for the U.S. job market or even worse, these American companies may be forced to move those jobs overseas. It is a well-known fact that young American women have a mostly negative opinion of computer science. Just 10% of college-bound high school girls have a "very good" opinion of computing compared to 45% of boys. At the college level, the number of women undergraduates majoring in computer science has dropped a staggering 80% over the past decade.[3]

The History of Women in Computing

The shortage of women in computing has never been because of low performance in computing or a lack of intelligence in technology. Women have played a vital role in computer pioneering from the very beginning. In the nineteenth century, Augustus Ada Byron Lovelace was a mathematician who collaborated with Charles Babbage on the Difference and Analytical Engines, which are regarded as the theoretical and practical precursors for the modern computer. Lovelace was best known for her 1843 translation from French to English of Luigi Menabrea's report on Babbage's Turin lecture; to which she added her own voluminous notes. Her paper discussed the Difference Engine, the first automatic calculating device, and the Analytical Engine, which contained the first set of principles for a general-purpose programmable computing machine. Lovelace's series of notes included a table describing the operations necessary for solving mathematical problems. She therefore became the first conceptual programmer for Babbage's Analytical Engine. In subsequent writings, she developed the "loop" and "subroutine" concepts a century before electronic computing machines appeared. These are two of the most important concepts used in computer programming.

Lovelace was a strong-willed, creative, intelligent, woman during the Victorian Era, when women in science were rare. Even so, her work was highly regarded by Babbage and the renowned mathematician Augustus DeMorgan, and she associated with intellectuals of her time, such as Michael Faraday, Charles Wheatstone, and John Herschel. The U.S. Department of Defense's high-level programming language, Ada, is named in honor of her contributions and pioneering spirit.[4]

Another pioneer in the field of digital computers was Grace Murray Hopper, a Vasser College and Yale University graduate. In the early 1940s Ms. Hopper joined the U.S. Navy and was assigned to a project on the Mark I at Harvard University with a Commander named Howard Aiken. While working on this project at Harvard she designed and implemented a program that computed the coefficients of the arctangent series. This introduced Ms. Hopper to the world of computer programming and she became what she referred to as, “the third programmer on the world’s first large-scale digital computer” which was named ENIAC. Ms. Hopper also coined the computer programming phrase “debugging” into the computing language which was the byproduct of a 1945 incident when a moth flew into a computer and caused a relay to fail while she and other programmers working on the Mark II project. That moth is now on display at the Naval Academy in Dahlgren with a note attached saying, “First actual case of bug being found”. Ms. Hopper is often referred to as the grandmother of COBOL, a computer programming language. In addition to all of her work in COBOL, she was a real technical visionary. She foresaw many applications for computing, including artificial intelligence, which she believed could replace the human brain. Ms. Hopper’s contributions are very well known and include ideas about tools and techniques of compiling and programming that are now commonplace: subroutines, translation of formulas, relative addressing, linking loaders, code optimization and symbolic manipulation.[5]

Facts and Myths of CS

Stereotypes of women in CS

Stereotyping in computer science has probably existed since computing began in the 1800s. Since the beginning stages women have been an integral part of computing and the technology industry. Their contributions include financing the first computer projects, designing the first computers and, because of the war effort during World War II, computer programmers were almost exclusively women. It might appear that with all these contributions to CS by women, the industry would be more sensitive to women rather than mistreating, stereotyping and isolating women. In the early days, thus may have been true, but eventually temperaments and attitudes changed. In fact, when early women programmers were asked how they were treated, most responded that they received equal treatment and were given the same respect as the men. These women felt it was not until the later years of computing that the computer science industry began unfair practices against the female workers. Women began to be stereotyped as good candidates for programming because, “programming requires lots of patience, persistence and a capacity detail and those are traits that many girls have”.[6] The cause of this change in perception of women in computing is believed to be as a result of the computer science industry transforming its business structure. The CS industry, as it began to expand into large hardware and software companies, adopted the business practices of male dominated corporations of that time.

Although this unequal treatment began to become a pervasive fact of American business, women continued to make important contributions to the CS industry. In the late 1950s the first real-time control computer and the first to use time-sharing was developed at MIT. Several women contributed to the development of this computer which was named the Whirlwind, including Judy Levenson (now Judy Clapp) who, at the time of the development of Whirlwind, Clapp had just received her M.S. in applied science from Harvard. Because of her knowledge and training she also helped to program a prototype of one of the first non-numerical applications of computers which was an air defense system using radar to track flying aircraft and direct the courses of other aircraft. When the operational version of this system needed to be programmed, several hundred people were hired and taught to program in assembly language. About 20% of these programmers were women and interestingly some of the best programmers ended up being those who majored in Music and English in college.

Dispelling the Stereotyping Myths

This Curriculum Unit (CU) is geared to addressing what may have been a less-than-promising future associated with CS for minority women. In doing so, the CU will seek to dispel many stereotypes which exist about CS by females and, more pointedly, by middle school and high school minority females. As explored earlier in this CU concerning the pioneers of computing, there was much excitement for women in the early days of computer programming and the initial design work associated with electronic computers and their programming languages. Women were a very large part of these groups who began this evolution of a new industry and, of which, grew out of this scientific field of study. It is my belief that women should continue to be an important part of the growth of the computer science field and industry as it works its way into more and more of every person's life, regardless of gender.

Women of Color Can Change the World with Computing

Minority women, in particular, have an opportunity to shape an industry and change our world in ways that have never been seen before in the history of the United States or any other nation. And with PC adoption in emerging markets around the globe growing so quickly, it is estimated there will be more than two billion PCs in use by 2015 according to Forrester Research. Therefore, whereas it took 27 years to reach the one billion mark, it will take only 7 years to grow from 1 billion to 2 billion. The opportunity for minority females is endless and would allow this group to achieve goals which are important to them such as helping society, while contributing to a better world.

Thus there is no reason, at least at the educational level, that women should be less prepared for these positions than men. A glaring statistic is the average starting salary for Computer Science grads with a BS is $60,000 a year. That is almost twice what the average teacher makes coming out of college with a Bachelor's degree in Education. But the high beginning salaries and wonderful opportunities for these CS grads seem to have made no impact on the minority female college students desiring to major in computing. The perception has been that CS doesn't deliver what girls are looking for in a career. They want work that is interesting, meaningful and inspires passion and as long as teenage minority females see CS as being boring, antisocial and lacking impact on their personal lives they will continue to look for other careers, no matter how much money these jobs pay.

According to a nationwide marketing study conducted with hundreds of high school girls by the Dotdiva.org group, the words these girls associate with computer science are "boring," "hard," and "nerd." This is a stereotype that continues to plague computer science and creates the ideal that only the smart misfits find their way into this field of study. At art schools such as NWSA that might not be as big of an issue simply because our students call themselves "misfits." In other schools, however, that can be a real deterrent to attracting minority females. Put in a more concise way, teen girls still perceive computer science careers as those which attract highly intelligent males who are, nevertheless, misfits and who inevitably work in boring jobs, sitting in front of computer screens all day, in order to solve mundane and irrelevant problems which never affect the "real" world. NWSA students do not have that same perception of CS because they believe being different, or a "misfit," is an advantage in life.

What Women Search for in Life

The worst part about this whole stereotype and misconception among young females is not that 'geeks' work in computer science. The worst part is that girls perceive having this poor image of computing careers as real and consequently believe that jobs in computer science might not allow them to live the lives they want and pursue what they value in life. The very thing that many young girls want to do, in both their lives and their careers, is to make a difference in the world. According to a nationwide survey, girls believe computer science fails to deliver the three most important characteristics they wanted out of their career:

1. Being passionate about their jobs (78%);
2. Doing interesting work (77%);
3. Having the power to do good and make a difference (56%).[7]

Women and Video Gaming

Another side of computing that is quite popular these days is the area of gaming. Nearly half of all gamers these days are female according to the Entertainment Software Association, the video game industry's trade group. Although stereotypes have a very negative effect on women in computing, the abuse women have received by gamers on the Internet may be even more detrimental to their interest in CS as a career. An example of this Internet abuse occurred in October 2014 when Anita Sarkeesian, a cultural critic, was to speak at Utah State University (USU). Gamers started a hashtag, #GamerGate, to campaign against what they perceived as ethical corruption among gaming journalists but recently it has evolved into a place to harass women like Sarkeesian for being a "social justice warrior" or *SJW*. The Internet has long been a place for people to seek anonymity so they can voice their views without repercussions for their beliefs and actions.

GamerGate came to national attention when Sarkeesian canceled her scheduled talk at USU because of death threats the school received against Sarkeesian, including one that promised "the deadliest school shooting in American history." The even greater threat Sarkeesian felt was that USU would not ban guns from inside the auditorium she was supposed to speak in. The school even refused to use metal detectors, but they told Sarkeesian they would check backpacks.

Sarkeesian is a vocal critic of the sexist depiction of women in video games. Sarkeesian and other women like her have been targeted because of their criticism of GamerGate and, according to a publication called Tech Times described GamerGate "a misogynistic online mob harassing female developers, journalists and even academic researchers."[8] Sarkeesian, a 31 year old female graduate of Cal State Northridge, illustrates examples of how video game developers use sexist language towards women in her video series "Tropes v Women". She also dissects 30 video games in her most recent episode on the website "Feminist Frequency" in a series called "Women as Background Decoration" including "Grand Theft Auto IV," "Super Mario Galaxy," and "Assassin's Creed.". She at looked at what she called "the subset of largely insignificant, non-playable female characters whose sexuality or victimhood is exploited as a way to infuse edgy, gritty or racy flavoring into game worlds." These women, she says, are "hollow shells with little to no personality" and "sexual playthings" who "allow the perpetuation of male violence."[9]

Although the critics have been very serious about condemning the threats of violence and harassment of these women it must do more. Minority females are not necessarily interested in working in an industry like video gaming where the women are depicted as almost subhuman. The demeaning aspects of games can only be eliminated, however, if men are willing to speak out, and if parents actually observe the games their children are playing on their computers and video gaming systems. An industry that denigrates women in the way that it does cannot expect to recruit women into its workforce and then not anticipate some backlash. The more this issue is exposed and revealed in a public forum the more our culture will condemn its' wrongful portrayal. CS and the gaming sector will see the need for change as it progressively continues to grow the industry by hiring and actively recruiting females. It will have no choice but to change because these women will begin to have an impact on changing the gaming world from the inside out.

Reversing the Negative Trend for Women in CS

It is my belief that teachers will help change the trend of women choosing other fields of study instead of CS. If each educator only changes a few students' perceptions of CS, this change has the ability to make a significant impact on our global community. All teachers can make a change; I committed to doing my part for this country, minorities, women and, most directly, the NWSA student population. The way I will do this is by recruiting and encouraging young, minority females to take a CS class then introduce computer programming in this entertaining and empathetic way. Other teachers can begin to do the same and in a short period of time our minority female students have changed an industry and, thus, their world.

Objectives

Computer Science can empower young girls to do 'good' in this world. It also allows girls to connect to their community, make a difference in other people's lives by reducing energy consumption, improving health care all over the globe, enhance everyone's own security, reduce the amount of pollution on the planet, benefit the downtrodden and enhance learning through alternative educational opportunities. These are issues which should entice girls to begin to explore the many opportunities afforded them in the field of computer science. Snap! can be their introduction to computing in the 21st century. With the Snap! Program software, minority female students have a chance to develop a new, exciting skill which can enhance their ability to create and discover new ideas in their education.

Introduction to Computer Programming
Since SNAP! is a computer programming language, I like to explain to my students that this visual programming language allows them a unique way of communicating with the computer so this machine can perform the functions they want it to complete. Unlike most programming languages, though, Snap! is a visual language. The difference that this gives a programmer is that instead of writing a program using the keyboard, the Snap! programmer uses the same drag-and-drop interface familiar to computer users. This Snap! Program can be taught by any middle school or high school teacher because it is so simple to teach and easy to comprehend. Since Snap! is a visual program it does not take an extremely long time to read about or to explain to students (or teachers).

Passing along knowledge of programming to younger children

As I reflected back on what I learned about programming this summer, I realized that this is something students of all ages and competencies, even ELL students, can learn to do in a short time period. N.C. State University actually brought in rising 9th grade students from Puerto Rico to help the teachers with learning this Snap! Program. Several of these student-teachers actually struggled, at times, with the English language, but Snap!, being a visual computer language, made the speaking language barrier a minor issue.
The N.C. State students, as well as the other college students the CS department at NCSU brought into the workshop as student teachers, even described a new volunteer program which the university is pioneering that takes Snap! Programming to local elementary school students. This volunteer program began in the Raleigh, NC area, but as a result of the success of this volunteer effort other universities around of the country, mostly surrounding these CS student's respective universities, are joining the programming movement. Elementary students are learning how to program with Snap! while

simultaneously learning from young females from local universities. It is not limited to college students, though. There is now a high school in Orlando, FL where the CS students go into their community elementary schools and pass along their knowledge about computer programming using Snap!.

The Programming Software

The SNAP! Software is a programming language that reminds students of Legos in shape and in the unique way the blocks fit or SNAP! together. The Snap! Program can help to recruit young, minority females to computing in a way and style of learning which helps dispel the ‘boring’ and disconnectedness most girls feel exists in CS today
Software Application:
When starting Snap! you will see the regions on the computer screen arranged at the top left with block tabs, below that is the Palette (a word familiar to art students) , to the right middle is the Script Window, above that window is the Sprite Tabs then in the largest and brightest section is the Stage. Below the stage is the Sprite Corral and above the Stage is a browser toolbar. Even the labeling and wording used in this programming language is student art friendly!

A Snap! program consists of one or more scripts, each of which is made up of blocks. A typical script consists of five blocks with three different colors, corresponding to three of eight palettes in which blocks can be found. See the example below:

The script area at the top left corner of the window shows one palette at a time, and the user chooses it with the eight buttons just above the palette area. In a typical script, the gold blocks are from the Control palette; the green block is from the pen palette.

The Pen writes when commanded to be down and stops writing when commanded to be up. The blue blocks are from the Motion palette. A Script is assembled by dragging blocks from a palette into the scripting area (Script Window) in the middle part of the window. Blocks snap together (hence the name Snap! for the language) when dragging a block so its indentation is near the tab of the one above it. When a block is dragged near another block a white horizontal line appears and acts like a magnet to snap the blocks together when you let go of the block you were dragging.

Teaching Strategies

When beginning a Snap! program it may be more convenient if the instructor projects the computer images onto a large screen so the students can slowly follow what is going on with the program and the assembly of the blocks. A through explanation of what you want to accomplish with your program is vital to keeping students interested in the lesson. This is where a working knowledge of computer programming, math and how to devise algorithms helps an instructor. It is not a necessity at all, but it does make it easier to perform higher function programs. (Refer to Appendix 2 for teacher references.)

Additional Programming Benefits for Students

I do need to interject a note here about the perception of most people, especially young women, about what computer programming is and why those with different approaches to "technology knowledge" and experience don't necessarily have a huge disadvantage in programming. One of the students that our N.C. State CS professor had helping us was a young female student majoring in Fisheries and Wildlife. She was minoring in CS only because she took an elective computer programming course and did very well despite her lack of any previous experience with computers. She said she did not really know how to use even MS Word well before she took her first CS course. With Snap!, she explained, basic computer programming began to make sense to her. This young lady stated that the visual aspect of the Snap! programming language it made it easier for her to understand how the program ran the command statements. She also believed CS had helped her organize her thoughts in a more logical way. She said after taking her first course in CS the process of writing research papers changed from a haphazard method before CS, to a well organized method with an outline used for structure. This now gave her a better idea about the purpose of the project before she began the writing process.

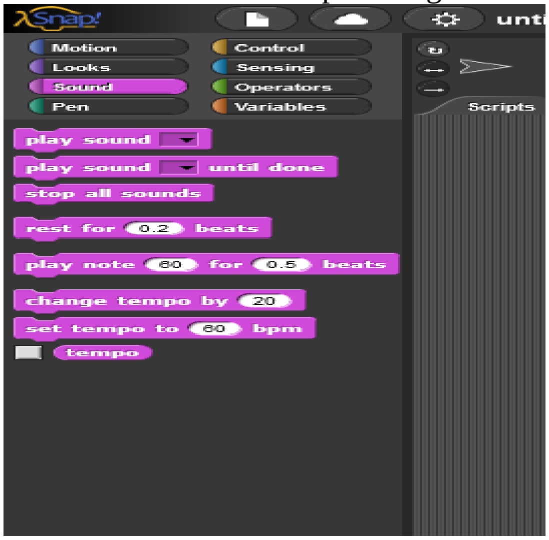
Your First Lessons

The first 2 simple lessons I encourage teachers to use in order to spark their students' interest are 'Make a sprite sing' and 'Meowing - One at a time or in Unison?" These lessons are especially appealing to young females students because it demonstrates a facet of software that is empathetic and not entirely "purpose-driven." The other reason is this will be good way to begin programming and help students understand how to organize in Snap!

The lesson 'Make a sprite sing' is written so the students can make a quick song. This gives students an opportunity to use the blocks in the 'Sound' tab as well as allows for students to change the default numbers so they can make the song play as long as they want it to play. It also is the student's first opportunity to figure out how to connect and disconnect the blocks, and how to remove a piece from inside a long script of the program or song. The lesson encourages the students to make many copies of one of the blocks in a row to hear the resulting sound. Then the students are told to try this for each block. The result is a very entertaining piece which students created by themselves using their creativity and at the same time they have "written a simple computer program with SNAP!

'Make a Sprite sing' Lesson

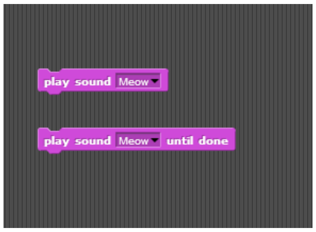
Below is what 'Make a Sprite sing' will look like in SNAP!



```
play sound
play sound until done
stop all sounds
rest for 0.2 beats
play note 60 for 0.5 beats
change tempo by 20
set tempo to 60 bpm
tempo
```

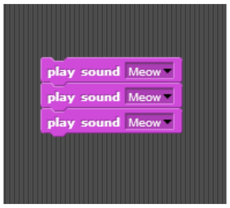
'Meowing - One at a time or in Unison?' Lesson

The other simple lesson 'Meowing - One at a time or in Unison?' starts to demonstrate to students how sprites and blocks interact and affect one another. For example, the "play sound" blocks from the 'Make a sprite sing' lesson allows students to control when and how they hear the sounds. Notice the difference in these two 'Sound' blocks?



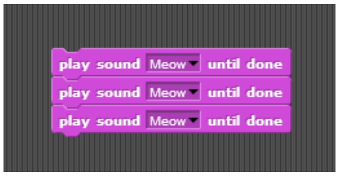
```
play sound Meow

play sound Meow until done
```

If you set up a small script like the one below, how many meows will you hear? The answer is one.



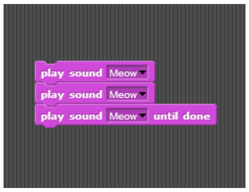
```
play sound Meow
play sound Meow
play sound Meow
```

What happens if you set the small script up like this below? How many meows do you hear? The answer is three consecutive “meows”.



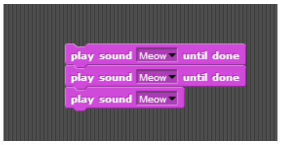
```
play sound Meow until done
play sound Meow until done
play sound Meow until done
```

Then you encourage students to experiment with these blocks. Have them try two ‘play sound (meow)’ blocks and then one ‘play sound (meow) until done’ block like this below. Students will only hear one “meow”. Call this #1 example.



```
play sound Meow
play sound Meow
play sound Meow until done
```

After doing that, tell students to try two ‘play sound (meow) until done’ blocks and then one ‘play sound (meow)’ block like this example below. Students will hear 3 “meows” consecutively. Call this #2 example.



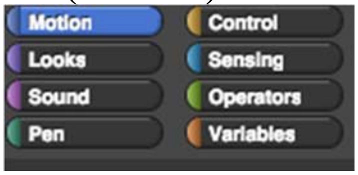
```
play sound Meow until done
play sound Meow until done
play sound Meow
```

Have the students explain the difference between #1 and #2 examples and why they heard a different amount of “meows”.

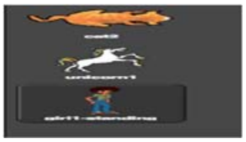
(Refer to Appendix2 -for Students and Teachers for student worksheets and first exercise in SNAP!)

Additional starting lingo for Snap!:

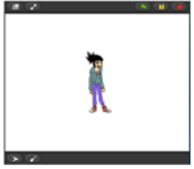
Tabs (for blocks):



Costumes (Each Sprite can have multiple costumes)



The Stage:



Conclusions

As you can see the Snap! computer programming language is geared towards students and relates to what they are familiar with in today’s technological age. The way to gain students’ attention is to be able to relate to what they know, gain their trust, and then begin the teaching process. The intuitive nature and interactive way in which Snap! performs makes it a “hit” with the intended target audience: minority females. It also introduces students to computers in a fun way, thus dispelling some of the old stereotypes about computing in the 21st century. The bottom line is that if the students aren’t afraid of technology and believe they can be successful, they are more apt to try CS courses in the future without any negative, preconceived notions. This is the purpose of a career in education, to make learning more enjoyable for students, thus leading to more fulfilling lives for all women, especially minority females.

Appendix 1: Teaching Standards

In this Curriculum Unit (CU), I cited specific textual evidence to support the analysis of technical texts. This unit provides an accurate summary of the text distinct from previous knowledge and cites opinions about the material from several sources. It also addresses and follows in an exacting way multistep procedures when carrying out or performing technical tasks for the SNAP! programming. This was done based on the Common Core standards CCSS.ELA-Literacy.RST.6-8.1, 8.2 and 8.3 for Grades 6-8.

The CU also analyzed the purpose in providing an explanation, describing a programming procedure, or discussed it in a text. This was very important to the success of the unit because we are performing a task in a visual programming language. If the procedure is not explained in text it would be difficult, yet not impossible to complete the computer program. By the end of the unit, students will have had the opportunity to perform and comprehend 21st century technology functions independently and exhibit these skills proficiently enough to write a simple program in SNAP! In doing so, the students will have achieved the Common Core Standards CCSS.ELA-Literacy.RST.6-8.5, 8.6 and 8.10 for Grades 6-8. In all the programming completed in SNAP! and this CU, I tried to give students some key ideas so they could craft and structure a visual computer program while integrating their previous knowledge and ideas in a way that conforms to the range of reading and the level of text complexity according to their grade level. I hope students discover that computing can be fun and an interactive learning process not an activity that is boring and dull.

Citations

(DoDEA 2014)
(George Lucas Educational Foundation 2014)
(TeachThought LLC 2014)
(ASCD 2012)

Appendix2 – for Students and Teachers

Intro to and directions for Downloading SNAP! – url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/snap/welcome-to-snap2.html?topic=berkeley_bjc/intro_new/1-introduction.topic&step=2&course=&_sm_byp=iVVN7Hj7fVfTFSsj
The website for students and teachers to download the SNAP! software so the exercises and teacher led instruction can be executed offline.

INTRODUCTION TO SNAP! - FIRST EXERCISE – url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/drawing/exploring-the-stage.html?topic=berkeley_bjc/intro_new/1-introduction.topic&step=7&course=&_sm_au_=iVV0MPfp5jVTSpvr
The website used to explain the first lesson students and teachers will execute in SNAP!

BLOWN TO BITS - Chapter 1 – url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/snap/welcome-to-snap2.html?src=http%3A%2F%2Fwww.bitsbook.com%2Fwp-content%2Fuploads%2F2008%2F12%2Fchapter1.pdf&topic=berkeley_bjc%2Fintro_new%2F1-introduction.topic&title=%20Blown%20To%20Bits%3A%20Ch&_sm_au_=iVV0MPfp5jVTSpvr
This website is used for explaining the Lesson Topic for the First Lesson in Snap!

Snap! Program and the Stage - url:http://snap.berkeley.edu/snapsource/snap.html?_sm_au_=iVV0MPfp5jVTSpvr
A website used to explain what the Stage in a SNAP! program is and how to identify tabs on the Stage

Appendix2 – *for Teachers*

Curriculum for SNAP! – url: http://bjc.berkeley.edu/
A website explaining the basic reason for SNAP! and why use and execute SNAP! in the 21st century classroom.

Introduction to SNAP! - First Exercise – url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/drawing/exploring-the-stage.html?topic=berkeley_bjc/intro_new/1-introduction.topic&step=7&course=&_sm_au_=iVV0MPfp5jVTSpvr
The website used to explain the first lesson students and teachers will execute in SNAP!

BLOWN TO BITS - Chapter 1– url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/snap/welcome-to-snap2.html?src=http%3A%2F%2Fwww.bitsbook.com%2Fwp-content%2Fuploads%2F2008%2F12%2Fchapter1.pdf&topic=berkeley_bjc%2Fintro_new%2F1-introduction.topic&title=%20Blown%20To%20Bits%3A%20Ch&_sm_au_=iVV0MPfp5jVTSpvr
This website is used for explaining the Lesson Topic for the First Lesson in Snap!

(1) The First Lesson in SNAP!

(1) Lesson Plan for how to create First SNAP! Program:

Subject: Computer Programming in SNAP!
Unit: ***Computer Programming is a SNAP! to Learn for Young Minority Females***
Lesson Topic: BLOWN TO BITS - Chapter 1; **BLOWN TO BITS - Chapter 1**– url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/snap/welcome-to-snap2.html?src=http%3A%2F%2Fwww.bitsbook.com%2Fwp-content%2Fuploads%2F2008%2F12%2Fchapter1.pdf&topic=berkeley_bjc%2Fintro_new%2F1-introduction.topic&title=%20Blown%20To%20Bits%3A%20Ch&_sm_au_=iVV0MPfp5jVTSpvr
Understanding Computer Programming in SNAP! - a visual programming language
Day: Day in 2014-2015 **Length of Lesson:** 90 mins.
Stage 1: Desired Results
Content Standard(s):
Common Core Standards CCSS.ELA-Literacy.RST.6-8.5, 8.6 and 8.10. These standards address and follow in an exacting way multistep procedures when carrying out or performing technical tasks such as SNAP! programming.
Understanding(s)/Goals
Students will understand:
The purpose and goal of computer programming in the 21st century.
What steps are required in the design and construction of a computer program.
How to write a computer program using the click and drag method using pictures and images instead of using the keyboard to write a computer program.
Essential Question(s):
What is the difference between SNAP! programming and the traditional method of computer programming?
What are the Main Sections of the SNAP! window? (Tool Bar, Palette, Scripting Area, Stage and Sprite Corral)
What are the 4 tabs used in SNAP! to "write" a simple program (MOTION, SOUND, PEN and SENSING)
Student Objectives (outcomes):
Write a SNAP! visual program using a method of programming which includes clicking, dragging and "snapping" the blocks together.

Stage 2: Assessment Evidence

Performance Task(s):

Complete the Blown to Bits - Chapter 1 directions and save the SNAP! program following the directions given at the website: SNAP! programming website– url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/snap/welcome-to-snap2.html?src=http%3A%2F%2Fwww.bitsbook.com%2Fwp-content%2Fuploads%2F2008%2F12%2Fchapter1.pdf&topic=berkeley_bjc%2Fintro_new%2F1-introduction.topic&title=%20Blown%20To%20Bits%3A%20Ch&_sm_au_=iVV0MPfp5jVTSpvr

Other Evidence:

Participate in the teacher directed interactive learning activity and complete the SNAP! program by following simple directions. Use this link: **BLOWN TO BITS - Chapter 1**– url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/snap/welcome-to-snap2.html?src=http%3A%2F%2Fwww.bitsbook.com%2Fwp-content%2Fuploads%2F2008%2F12%2Fchapter1.pdf&topic=berkeley_bjc%2Fintro_new%2F1-introduction.topic&title=%20Blown%20To%20Bits%3A%20Ch&_sm_au_=iVV0MPfp5jVTSpvr

Stage 3: Learning Plan

Learning Activities:

1. Complete and run your SNAP! Program
2. Save your project using the directions on the website

How did I differentiate my lesson?

Use classroom discussion, group collaboration and individual classwork to enable diverse learning styles to be expressed by certain students do a collaborative presentation with another student so lower performing students are not overwhelmed by SNAP!

(2) The Second Lesson uses PowerPoint

(2) Lesson Plan for the History of Women in Computer Science (CS)

Subject: The Impact Women have made to the field of CS

Unit: ***Computer Programming is a SNAP! to Learn for Young Minority Females***

Lesson Topic: The History of Women in Computer Science

Day: Day in 2014-2015 **Length of Lesson:** 90 mins.

Stage 1: Desired Results

Content Standard(s):

Common Core Standards CCSS.ELA-Literacy.RST.6-8.5, 8.6 and 8.10. These standards address and follow in an exacting way multistep procedures when carrying out or performing technical tasks in researching, organizing and completing a presentation in MS Office Powerpoint software.

Understanding(s)/Goals

Students will understand:

The importance and contributions of women to the field of Computer Science from its inception.

The leading women who help pioneer computing and what their impact was to CS.

The steps needed to research, organize, design and master a Powerpoint Presentation.

Essential Question(s):
Who were the most important women to the field of CS and what were their most impactful contributions to computing?
Why have their contributions not been highlighted and acknowledged by the Computer Science community and exhibited in more of the 21st century educators' classrooms?
Student Objectives (outcomes):
Develop and execute the skills needed to design a well developed PowerPoint Presentation.
Conduct research and gain insightful knowledge about Women Pioneers in the field of CS.
Stage 2: Assessment Evidence
Performance Task(s): Complete the PowerPoint Presentation using knowledge from internet research about Women who shaped computing.
Other Evidence: Students will conduct a five minute PPT presentation to the class to demonstrate their mastery of this material.
Participate in the teacher directed interactive learning activity and complete the SNAP! program by following simple directions.
Stage 3: Learning Plan
Learning Activities:
1. Complete the PPT presentation
2. Students will ask their classmates 4 questions about their PPT presentation that highlights the key points of the PPT.
How did I differentiate my lesson?
Use classroom discussions, group collaboration or individual research to enable diverse learning styles to be expressed by certain students and allow some students to do a collaborative PPT presentation with another student so lower performing students are not *overwhelmed by the workload. This needs to be teacher approved and supervised closely.*

(3) The Third Lesson uses SNAP!

(3) ***Lesson Plan for*** ***EXPLORING THE STAGE IS A SNAP! : POSITION ON THE STAGE***
Subject: Understand the Snap! Stage, discover what a sprite is and how to move it on the Stage
Unit: ***Computer Programming is a SNAP! to Learn for Young Minority Females***
Lesson Topic: The Stage and how to move/position a sprite.
Day: Day in 2014-2015 **Length of Lesson:** 90 mins.
Stage 1: Desired Results
Content Standard(s):
Common Core Standards CCSS.ELA-Literacy.RST.6-8.5, 8.6 and 8.10. These standards address and follow in an exacting way multistep procedures when carrying out or performing technical tasks when executing a movement on the stage of a visual computer program called SNAP!

Understanding(s)/Goals

Students will understand:

What a stage is used for in SNAP!
What a sprite is in SNAP!
How to position a Sprite on the stage and execute a movement of the sprite using simple math.

Essential Question(s):

What are the two positions or coordinates a sprite occupies on the Stage? (Horizontal and Vertical positions)
How is his position expressed in the horizontal and vertical positions? (x,y)
What is the center or origin of the sprite on the stage and how is it expressed with coordinates (0,0)

Student Objectives (outcomes):

The MOTION and PEN tabs will be used to move the Sprite and students will learn how to execute parts of a computer program using the SNAP! programming method to draw on the Stage with the Sprite.

Stage 2: Assessment Evidence

Performance Task(s): Complete the teacher directed exercise using SNAP! called *Experimenting with Drawing Commands.*

Other Evidence:

Participate in the teacher directed interactive learning activity and complete the SNAP! program by following simple directions. Start with this link: Position on Stage – url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/drawing/exploring-the-stage.html?topic=berkeley_bjc/intro_new/1-introduction.topic&step=7&course=

Stage 3: Learning Plan

Learning Activities:

1. Complete the activity using the MOTION and PEN Tabs to move the Sprite in Snap!
2. Students will follow and complete the teacher directed activity *Experimenting with Drawing Commands.*

How did I differentiate my lesson?

Use classroom discussions, group collaboration or individual research to enable diverse learning styles to be expressed by certain students and allow some students to try the FOLLOW THAT MOUSE activity. This does not need to be teacher approved and supervised closely. Students will begin to follow the lessons before and after Exploring the Stage: Position on the Stage. – url: http://bjc.eecs.berkeley.edu/bjc-r/cur/programming/intro/drawing/exploring-the-stage.html?topic=berkeley_bjc/intro_new/1-introduction.topic&step=7&course=

Bibliography

© 2010 WGBH Educational Foundation. *What's the Problem with Girls and Computing?* 2010. http://www.dotdiva.org/educators/problem.html (accessed June 20, 2014). Excellent source for information about what young female minorities perceive CS and computing to be in the 21st century and some stategies on how to reshape this negative perception.

Abcarian, Robin. "LA Times Columnist." *L.A. Times.* website. L.A. Times. Los Angeles, CA: LA Times website, October 17, 2014. This was used for information about how females are being treated by gamers and even educators as well as statistics concerning the number of female gamers in the world today.

ASCD. In *The Core Six: Essential Strategies for Achieving Excellence with the Common Core*, by R. THOMAS DEWING and MATTHEW J. PERINI HARVEY F. SILVER, pp. 40–44. Alexandria, VA: Association for Supervision & Curriculum Development, 2012. A reference used to help integrate teaching standards and strategies related to Common Core Standards.
DoDEA. *Common Core State Standards Initiative* . 2014. http://www.corestandards.org/ELA-Literacy/RST/6-8/ (accessed 2014). This site was used for information and direct Common Core Standards cited in this CU.

Gartner. *worldometers.* Worldometers. October 20, 2014. www.worldometers.info/computers/ (accessed 2014). Used for information and statistics about numbers of computers in the world and future statistics about 21st century computing around the world.

George Lucas Educational Foundation. *Edutopia - What works in Education.* 2014. http://www.edutopia.org/blog/ccia-10-visual-literacy-strategies-todd-finley (accessed September 29, 2014). Used to develop and gain insight into how strategies and Common Core standards can be used in the classroom and integrated in the 21st century classroom.

Gurer, Denise. "Pioneering Women in Computer Science." Edited by ACM. *SIGCSE Bulletin* (Communications of the ACM) 34, no. 2 (June 2002): 45-54. A source used for the history of women in Computer Science from the first beginnings of computing to present day in the US and around the world.

Landivar, Liana Christian. *Disparities in STEM Employment by Sex, Race, and Hispanic Origin.* Edited by American Community Survey. Economics and Statistics Administration U.S. Dept. of Commerce. September 2013. www.census.gov/prod/2013pubs/acs-24.pdf (accessed September 2013). This source was

used for current statistics and general information about computing for young females in the U.S. as well as trends about minority females working CS today.
Lifelong Kindergarten Group at the MIT Media Lab. *SNAP! (Build Your Own Blocks) 4.0.* 2014. http://snap.berkeley.edu/ (accessed June 16, 2014). This is the website to use for all SNAP! lessons, exercises and downloads needed for classroom applications for both students and teachers.

NCWIT. "By the Numbers." *NC WIT.* NCWIT. December 2011. www.ncwit.org (accessed October 2014). This source was used to gain knowledge, through in-depth studies conducted by several sources, using various methods concerning what young girls perceptions are about CS and computing.

TeachThought LLC. *2 Simple Strategies To Bring Technology To The Common Core.* 2014. http://www.teachthought.com/technology/merging-technology-common-core/ (accessed October 2014). This is an excellent source used to relate content to Common Core Standards used for teaching technology and how it relates to standards and teaching strategies.

Notes

[1] US Dept. of Labor, Bureau of Labor Statistics, 2011 (Occupational Category: 15-0000)
[2] Boardroom Insiders, 2012; Includes new and replacement jobs and assumes current undergraduate degree (CIP 11) production levels persist.
[3] Higher Education Research Institute (HERI), *College Freshmen Survey*, 2000-2009.
[4] Denise Gurer (1995) “Pioneering Women in Computer Science” Communications of the ACM. 38(1), pp. 45-54.
[5] Denise Gurer (1995) “Pioneering Women in Computer Science” Communications of the ACM. 38(1), pp. 45-54.
[6] Seigsohn, I.J., *Your Career in Computer Programming*, Julian Messner, Ed., Simon and Schuster, 1967. (Landivar 2013) (Abcarian 2014) (Gurer 2002) (NCWIT 2011)
[7] (© 2010 WGBH Educational Foundation 2010)
[8] LA Times by Robin Abcarian, *Anita Sarkeesian bravely confronts sexist video gaming culture,* October 17, 2014 ed.
[9] LA Times by Robin Abcarian, *Anita Sarkeesian bravely confronts sexist video gaming culture,* October 17, 2014 ed.