Computing Lessons in High School Geometry: A Method for Narrowing Gender and Racial Gaps in AP Computer Science

Gary Stark
Greeley Central High School, Colorado
August 5, 2015

NSF Grant #1262031

The Need

CASPER's Summer Research Experiences for Teachers Program is sponsored by the National Science Foundation through grant # 1262031, CASPER (Center for Astrophysics, Space Physics, and Engineering Research), Baylor University, the School of Engineering (Mechanical and Electrical) and the College of Arts and Sciences (Physics and Math). The RETs for the summer of 2015 were charged with the tasks of:

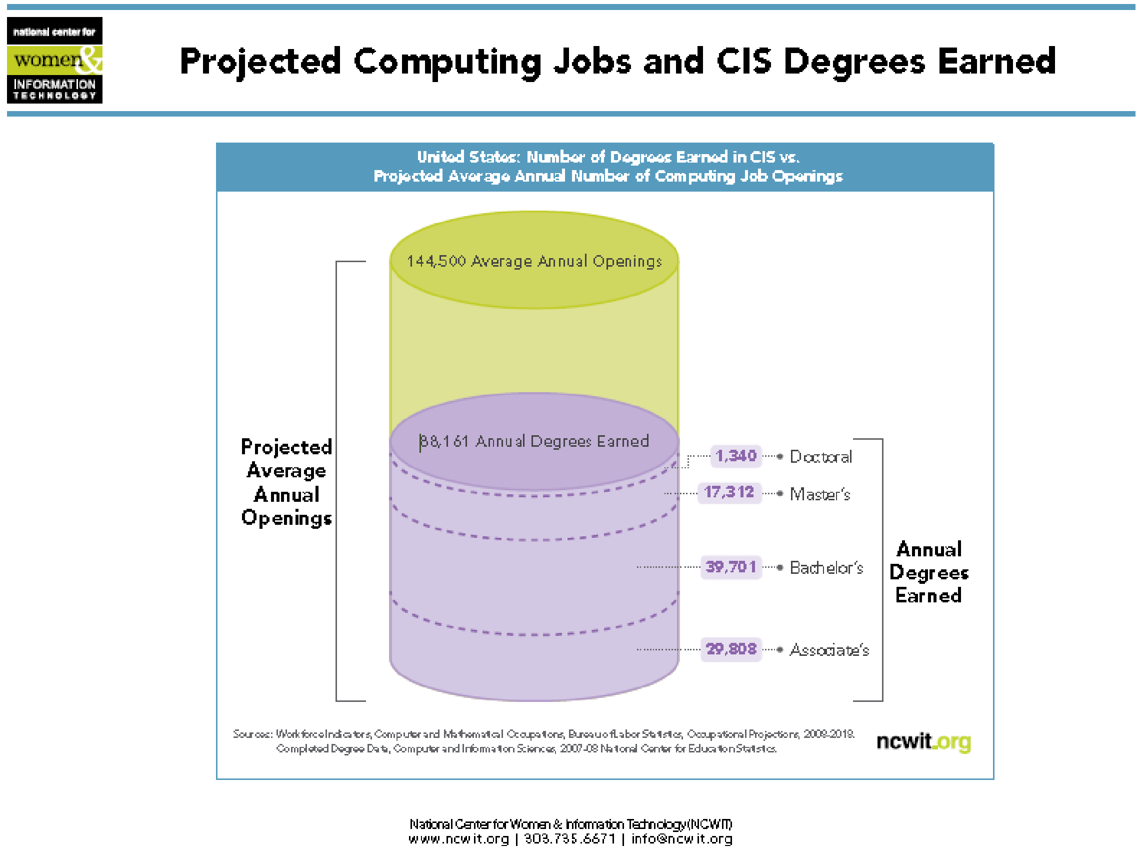
- Creating lesson plans that allow teachers to integrate a virtual programming environment as mini sessions during the normal class time (not as an afterschool program) (based on STEM+C grant application) for 6^{th}-8^{th} grades
- Reviewing and summarizing literature on Computational Thinking Skills
- Facilitating a CASPER evaluation project on a teacher summer PD

The STEM + Computing Partnerships grant being considered is program solicitation NSF 15-537. The researchers read through the grant solicitation, background information on the purpose of the grant solicitation, and research on how students learn computing and the current state of computing education. The highlights and summary of the research follow.

The US is not producing enough graduates majoring in computer science fields to keep up with the projected number of computing jobs. Using data from the Bureau of Labor and Statistics and the National Center of Education Statistics, the US will have 144,500 new openings in computer science fields annually through 2018, but only 88,161 degree in computer fields are awarded annually, leaving an average deficiency of over 55,000 trained workers (National Center for Women and Information Technology). Fig. 1A. shows the breakdown by degree type.

According to the Taulbee survey, over 60% of the graduate degrees in Computer Science and Computer Engineering were awarded to non-resident aliens and the proportion of Ph.D. graduates who were reported taking positions outside of North America, among those whose employment is known, rose to 9.4 percent. There is a push to increase the number of students majoring in computing fields, but colleges will have a tough time preparing for a potential increase in undergraduates as only 27.3 percent of 2013-14 (Ph.D.) graduates in the computing fields took North American academic jobs, an all-time low since the data began being tracked in 1989-90 (Zweben & Bizot, 2015). The impact of finding qualified instructors of undergraduates in computer science courses is already being felt. The Association for Computing Machinery Education Council's Diversity Task Force has focused much discussion on the enrollment boom. "A dramatic rise in enrollment at many universities is leading to limits and caps that are likely to reduce diversity in undergraduate computer science programs. We estimate that there may be 1000 CS teaching positions available in the current market, but it's likely that less than half will be filled." (Timanovsky).

Having students take AP Computer Science in high school increases their chances of majoring in computer science in college. According to the College Board, students who take AP Computer Science in high school are eight times more likely to major in computer science. In the summer of 2010, Google surveyed a sample of its US employees. 98% of the employees who majored in computer science reported having exposure to computer science before college. Only 45% of the non-computer science majors reported having exposure to computer science. Unfortunately, the number of students taking AP Computer Science has not been improving in comparison to the other STEM AP courses. In 2013, there were 365,868 AP Calculus exams taken, compared to only 29,555 AP Computer Science exams. Fig. 1B. shows the recent trend for STEM AP exams through 2011.



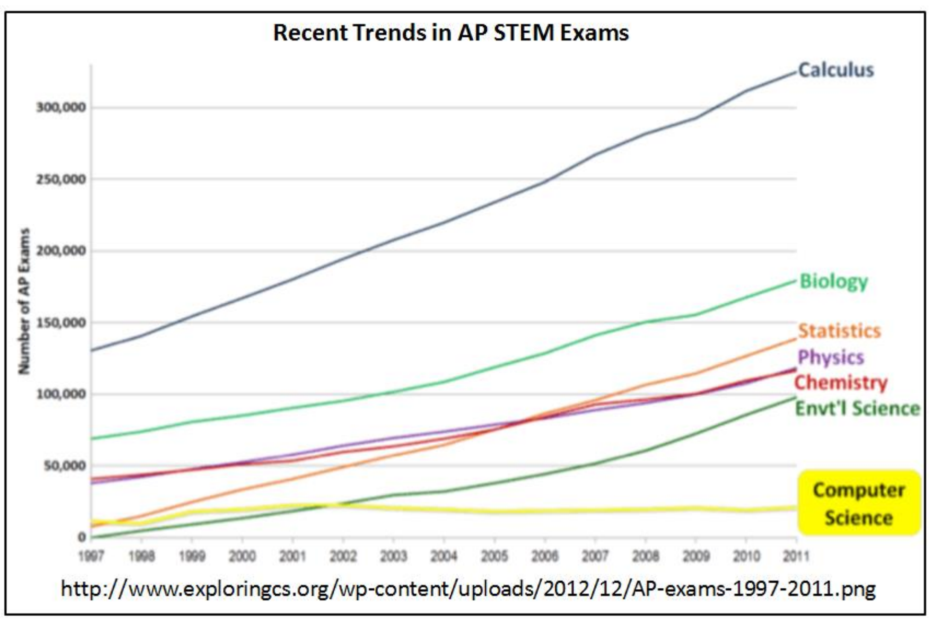


Fig. 1A. The projected average number of annual computer related job openings in the United States and the annual number of degrees earned. Fig. 1B. The number of AP exams taken in STEM courses from 1997-2011.

There is hope that more college students will become interested in majoring in the computing fields as the National Science Foundation and the College Board are preparing a new advanced placement computer science course titled Advanced Placement (AP®) Computer Science Principles. Entry into the new course will hopefully not seem as such a major step for students new to computing. Students performing at a satisfactory level may receive college credit for the course, which would help the burden of colleges teaching a potentially increasing number of undergraduate computing majors. In 2010, NSF began the CS10K initiative, which seeks to have 10,000 well-trained computer science teachers in 10,000 US high schools before the roll-out of the AP CSP course in the 2016-17 school year.

The participation of women and minorities in the computer science field is significantly low. According to the 2014 Taulbee Survey, only 14.7 percent of the Bachelor's Degrees in the computing fields, computer science, computer engineering or information, were awarded to females during the 2013-14 school year, though females make up about 50 percent of the population and earn 57.2% of the total Bachelor's Degrees in the US (National Center for Education Statistics). According to the College Board, the percentage of AP Computer Science test takers that were female was only 19%. The percentage of females taking AP exams in other STEM courses is significantly higher. Fig. 2.

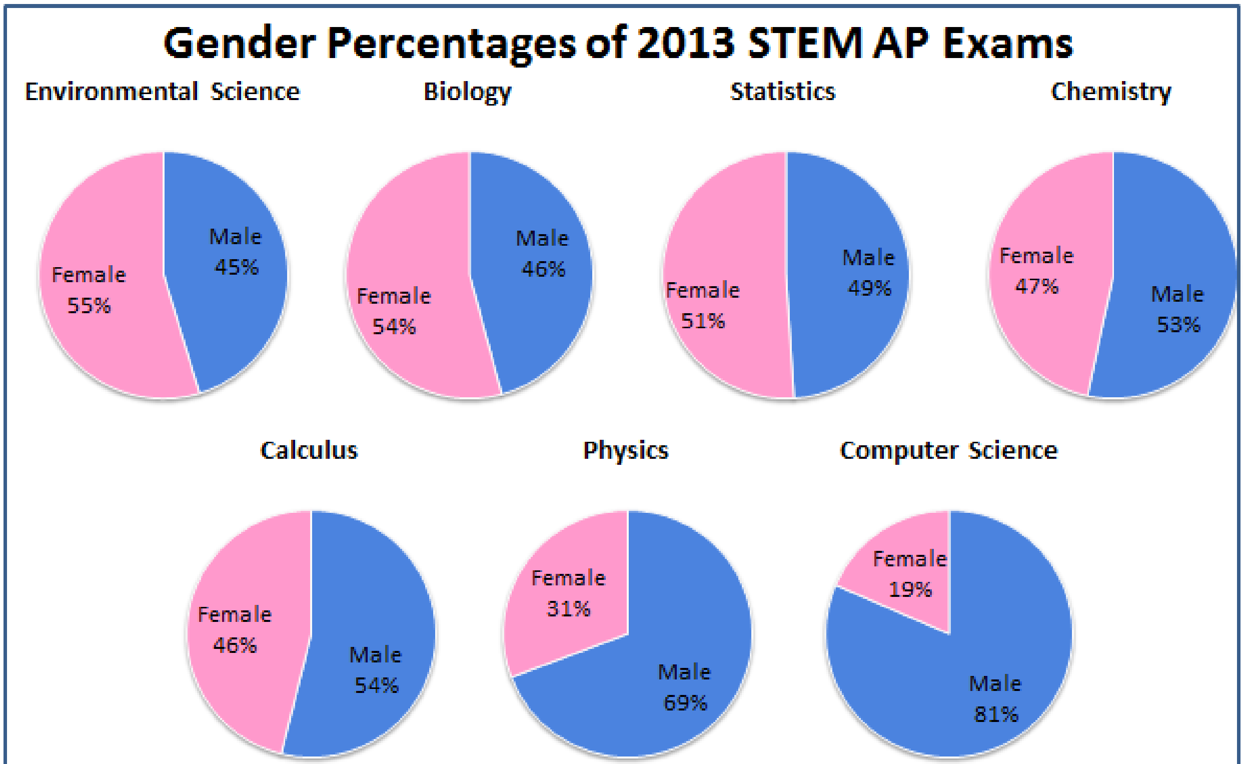
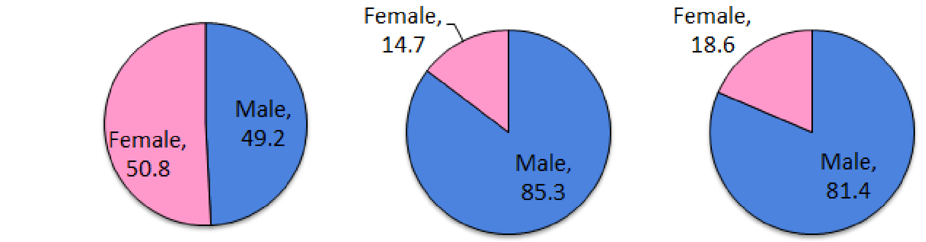


Fig. 2. The percentages by gender of 2013 AP exams in the STEM fields. Calculus includes AB and BC exams. Physics includes all B and C exams.



	US Census Bureau 2013 Population Estimates	2014 Taulbee Survey Bachelor's Degrees in Computer Sciences	College Board AP Computer Science Exam Takers
Female	50.8%	14.7%	18.6%
Hispanic or Latino	17.1%	7.7%	8.1%
Black or African American	13.2%	4.1%	3.7%

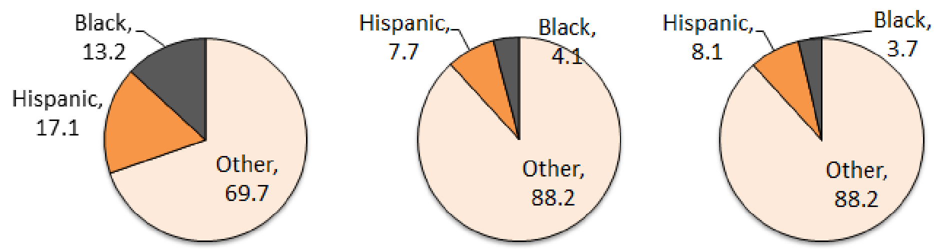


Fig. 3. Partial demographic breakdown of the following: US population estimates in 2013; Bachelor's Degrees earned in in 2013-14 in Computer Science, Computer Engineering, or Information; and the number of AP Computer Science exams taken in 2013.

Shocking statistics on the number of 2013 AP Computer Science exam takers were detailed by Barbara Ericson, including:

- No females took the exam in Mississippi, Montana, and Wyoming.
- 11 states had no Black students take the exam: Alaska, Idaho, Kansas, Maine, Mississippi, Montana, Nebraska, New Mexico, North Dakota, Utah, and Wyoming.
- 8 states had no Hispanic students take the exam: Alaska, Idaho, Kansas, Mississippi, Montana, Nebraska, North Dakota, and Wyoming

In an Education Week follow-up article, Deborah Davis, spokeswoman for the College Board, wrote in an email, "We were not surprised by Barbara Ericson's findings because unfortunately, computing courses have historically been dominated by white, male students." And later in the article: “One complicating factor, Ericson said, is that AP computer science courses "are more prevalent in suburban and private schools than in urban, poor schools." About 2,300 high schools are officially recognized by the College Board as offering AP computer science for 2013-14—a fraction of the 15,000 high schools that offer some type of AP course, she noted.” In a comment from the article, Mark Guzdial, professor in the College of Computing at Georgia Institute of Technology, wrote “I asked Barbara if she knew what was going on in Mississippi, and she told me something I was amazed at: There are no AP CS teachers in *any* high schools in Mississippi! There are only two “schools” that passed the AP audit in Mississippi, and both are on-line-only virtual high schools”. Again, this is an indication that access to many upper-level courses is not available to all students, quite often the disadvantaged students.

As stated earlier, the percentage of students majoring in computer science is related to their exposure before college and whether or not they took AP Computer Science. So if students, especially females, Hispanics, or African Americans, do not have exposure to computing in their schools, the chances are slim that they will major in computer science. The National Science Foundation is hoping that the new AP Computer Science Principles course will make the transition into computing less frightening. If the CS10K project is successful at training 10,000 computer science teachers for 10,000 schools, the majority of high schools would still be without trained computer science teachers. The total number of high schools is over 30,000 and some sources list the number over 40,000. When US News and World Report conducted its 2015 Best High Schools Rankings, it started with 29,070 public high schools (not counting the many private schools) and eliminated many from consideration, mostly because of smaller enrollment numbers. It reduced the number to 19,753 public high schools to be eligible for the rankings. The CS10K project would still leave over half of all schools without a trained computer science instructor. As stated by Janice Cuny, NSF's Program Director of Computer Education in her article "Transforming High School Computing: A Call to Action" in ACM's Inroads, "Minorities are more likely to be attending low-resourced schools that do not offer opportunities for high school computing." A successful CS10K project, though an excellent start at alleviating the shortage of computer science majors, will probably leave the majority of minorities without access to computing classes in high school.

Where can all students be given a chance to experience computing before college? Work can begin in middle school or earlier, but then more teachers would need to be trained. Again, quoting Janice Cuny's Inroads article, "The Directorate for Computer and Information Science and Engineering (CISE) at the National Science Foundation (NSF) supports efforts to improve computing education across the academic pipeline, in both formal and informal educational settings, in computing courses and as topics infused across other disciplines, and targeting both students and their teachers. The computing community needs to make progress on all of these fronts, but progress at the high school level is key".

The Lessons

One possible route for ensuring that all students have access to computing would be to offer small doses in geometry classes. Every high school offers geometry and just about every student, except those accelerated a few grade levels, takes it in high school. Included in this paper are six lessons involving transformations and two project frameworks for teachers who use Project Based Learning (PBL); one for the unit on transformations and the other for the trigonometry unit. The six lessons can be used if the classroom is traditional or uses PBL. Transformations are included in the standards for the Common Core and Texas TEKS for both 8^{th} grade and high school geometry.

The first five lesson plans take 15 to 20 minutes of class time while the sixth lesson takes about 45 minutes and could be done as a jigsaw. A three day introduction to the Scratch programming language is recommended before the lessons. Recommendations would be to follow most examples at the beginning of Majed Marji's book *Learn to Program with Scratch – A Visual Introduction to Programming with Games, Art, Science, and Math* through page 33. Students should at least program the Pong game from pages 15-19. After this introduction, many students will explore on their own outside of class. They will be ones to assist students throughout the lessons and will have a very good chance of taking AP Computer Science Principles or other courses with coding offered at their school.

The first day of the unit could be the second day of the course. The project statement for the first unit could be "How can we as game designers create a Scratch game to help middle school students learn about transformations including dilations?" The project launch could be demonstrating a Scratch game, such as the Star Wars game by user atomicmagicnumber at https://scratch.mit.edu/projects/42217958/. This will show students the level of programming they could be capable of in a few weeks. The teacher will start a "Need to Know" list for the content and logistics of the project. Teachers should also have a calendar and rubric for the project. The author will have these ready for use by the middle of August, 2015.

After three days of introduction to programming in Scratch, the transformation lessons can begin. The sections covered from the book include:

Chapter 1 Getting Started:

- What is Scratch p. 1-3
- Scratch Programming Environment p. 3-13
- Point Editor p. 13-15
- Your First Scratch Game (Pong) p. 15-19
- Scratch Blocks: An Overview p. 20-21
- Arithmetic Operations and Functions p. 21-22

Chapter 2 Motion and Drawings

- Using Motion Commands p. 25-30
- Pen Commands and Easy Draw p. 31-32
- The Power of Repeat p. 33-35

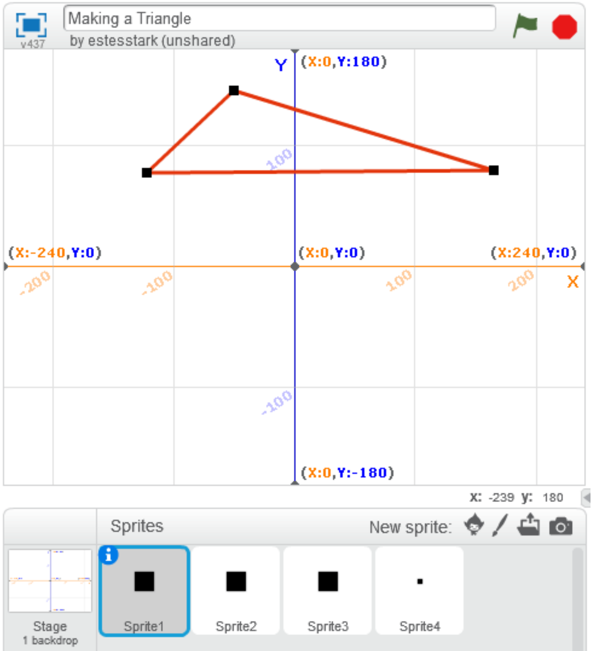
Possible Schedule:

- Day 1: Introduction, syllabus, interest survey
- Day 2: Project Launch
- Day 3: p. 1-15
- Day 4: p. 15-19 Students create Pong game
- Day 5: p. 21-35
- Day 6: Lesson 1 – Making a Triangle
- Day 7: Lesson 2 - Reflection Across the x-axis
- Day 8: Lesson 3 - Reflection Across the y-axis and Origin
- Day 9: Lesson 4 - Reflection Across the lines $y = -x$ and $y = x$
- Day 10: Lesson 5 – Translations
- Day 11: Lesson 6 – Rotations (This lesson could be completed using a Jigsaw.)

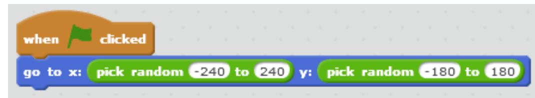
Lesson 1 - Making a Triangle

Name____________________________ Date_______ Block__

In the Scratch program “Making a Triangle”, the Stage (backdrop) and Sprite 1 appear as:

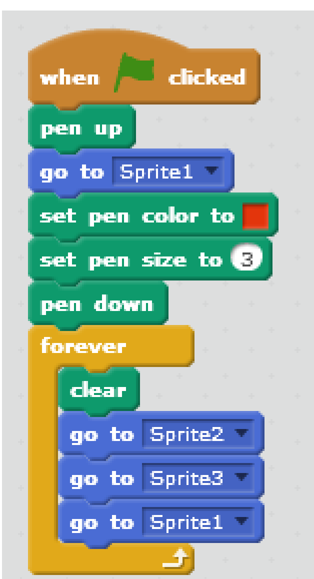


1. What does the script for Sprite 1 accomplish?



The scripts for Sprites 2 and 3 are similar.

The script for Sprite 4 is:



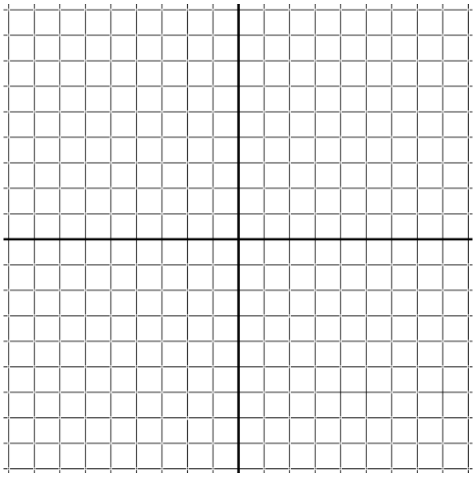
2. Why does this script draw a triangle?

3. If you click on a vertex (Sprite) and drag it to a new position, the triangle changes shape. Why does this happen?

Lesson 2 - Reflection Across the *x*-axis

Name_________________________ **Date**_______ **Block**__

Materials needed: colored pencils, straightedges

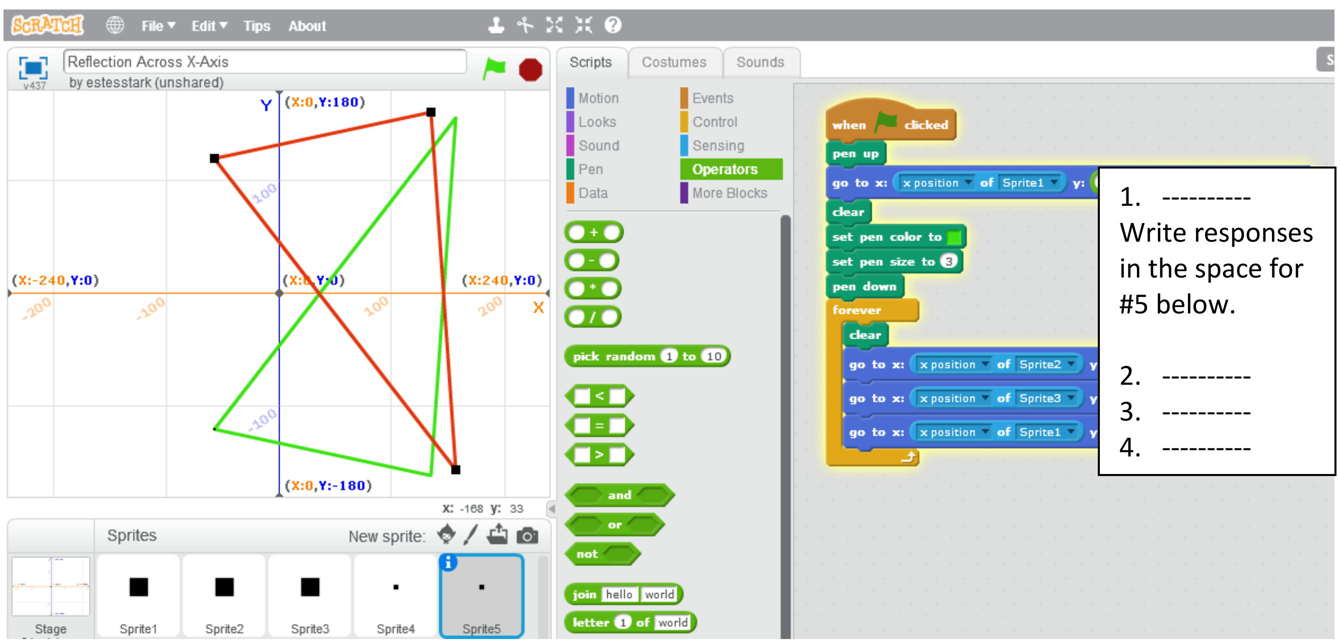


1. Use a red pencil. Plot 3 lattice points in the 1st quadrant and connect with straightedge to form triangle ABC. Reflect the triangle across the *x*-axis and form triangle A'B'C'. Label the vertices of both triangles with letters and coordinates.

2. With a green pencil, do the same with 2 points in the 2nd quadrant and 1 point in the 1st quadrant. Reflect across the *x*-axis and label with D, E, and F, primes (') and coordinates.

3. With a blue pencil, draw the same with a point in each of quadrants I, II, and III. Label with X, Y, and Z, primes, and coordinates.

4. Make a generalization for reflecting the point (*x*, *y*) across the *x*-axis.



5. A fifth Sprite and script have been added to the program from the previous lesson. What combination of operator and sensing code need to be added for each line for the *y*-coordinates?

1.

2.

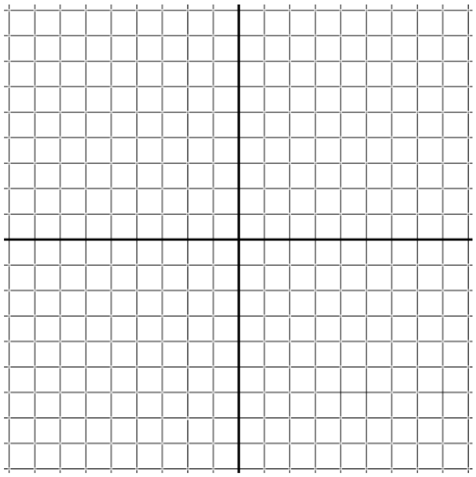
3.

4.

Lesson 3 - Reflection Across the *y*-axis and Origin

Name________________________ Date______ Block__

Materials needed: colored pencils, straightedges



1. Use a red pencil. Plot 3 lattice points in the 1st quadrant and connect with straightedge to form triangle ABC. Reflect the triangle across the y-axis and form triangle A'B'C'. Label the vertices of both triangles with letters and coordinates.

2. With a green pencil, do the same with 2 points in the 2nd quadrant and 1 point in the 1st quadrant. Reflect across the *y*-axis and label with D, E, and F, primes (') and coordinates.

3. Make a generalization for reflecting the point (*x*, *y*) across the *y*-axis.

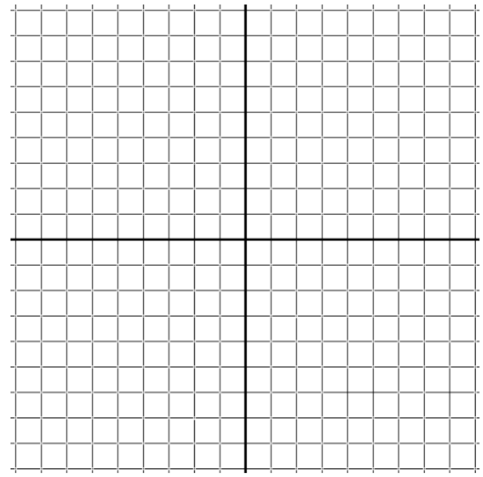
4. List the changes that need to be made to reflect the triangle across the *y*-axis.

a.

b.

c.

d.



5. Use a red pencil. Plot 3 lattice points in the 1st quadrant and connect with straightedge to form triangle ABC. Reflect the triangle across the origin and form triangle A'B'C'. Label the vertices of both triangles with letters and coordinates.

6. With a green pencil, do the same with 2 points in the 2nd quadrant and 1 point in the 1st quadrant. Reflect across the origin and label with D, E, and F, primes (') and coordinates.

7. Make a generalization for reflecting the point (*x*, *y*) across the *y*-axis.

8. List the changes that need to be made to reflect the triangle across the origin.

a.

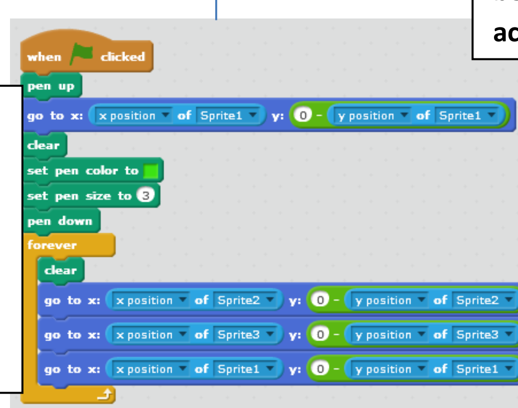
b.

c.

d.

a.

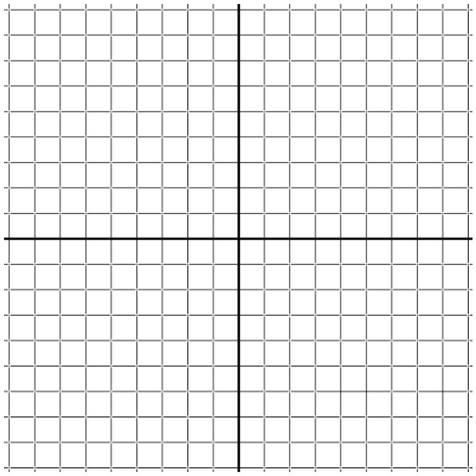
b.
c.
d.



Lesson 4 - Reflection Across the Lines *y* = -*x* and *y* = *x*

Name______________________ Date_____ Block__

Materials needed: colored pencils, straightedges



1. Use a red pencil. Plot 3 lattice points in the 1st quadrant and connect with straightedge to form triangle ABC. Reflect the triangle across the line $y = -x$ and form triangle A'B'C'. Label the vertices of both triangles with letters and coordinates.

2. With a green pencil, do the same with 2 points in the 2nd quadrant and 1 point in the 1st quadrant. Reflect across the *y*-axis and label with D, E, and F, primes (') and coordinates.

3. Make a generalization for reflecting the point (x, y) across the line $y = -x$.

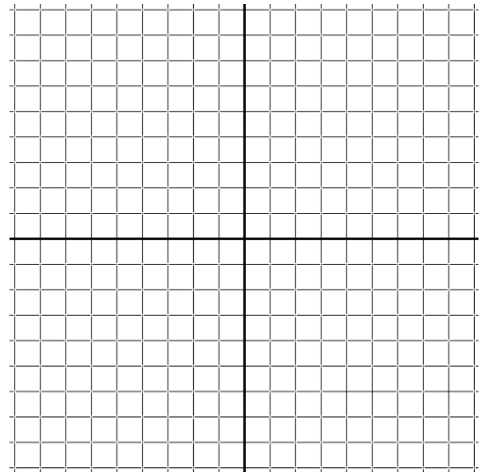
4. List the changes that need to be made to reflect the triangle across the line $y = -x$.

a.

b.

c.

d.



5. Use a red pencil. Plot 3 lattice points in the 1st quadrant and connect with straightedge to form triangle ABC. Reflect the triangle across the line $y = x$ and form triangle A'B'C'. Label the vertices of both triangles with letters and coordinates.

6. With a green pencil, do the same with 2 points in the 2nd quadrant and 1 point in the 1st quadrant. Reflect across the origin and label with D, E, and F, primes (') and coordinates.

7. Make a generalization for reflecting the point (x, y) across the line $y = x$.

8. List the changes that need to be made to reflect the triangle across the line $y = x$.

a.

b.

c.

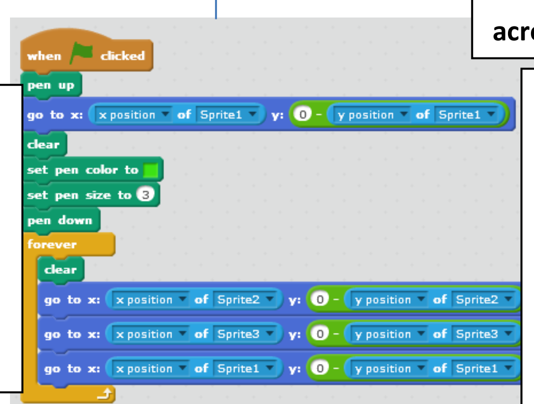
d.

a.

b.

c.

d.

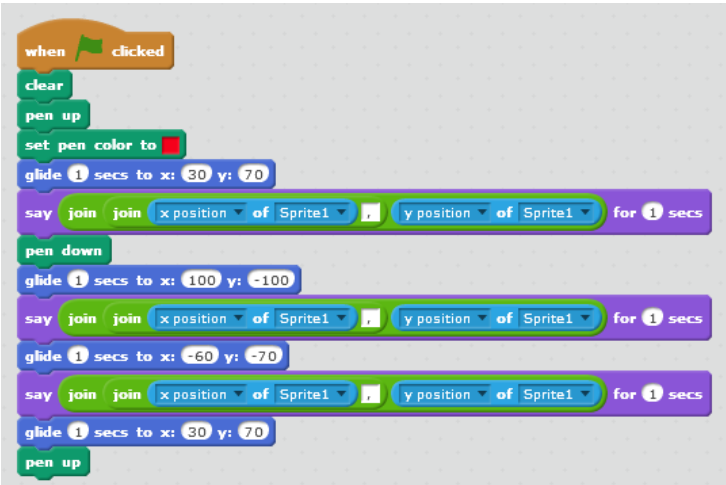


Lesson 5 – Translations

Name__________________________ Date_______ Block__

Run the Scratch program "Simple Triangle Translator" https://scratch.mit.edu/projects/69005220/#editor

This program contains new coding statements. Sometimes the statements are intuitive (make common sense), but other times you will need to ask others, check the index in a good programming book such as Learn to Program Using Scratch by Majed Marji, or you can search on the Scratch web page which will take you to the Scratch wiki. The new ones today will be partially explained here.

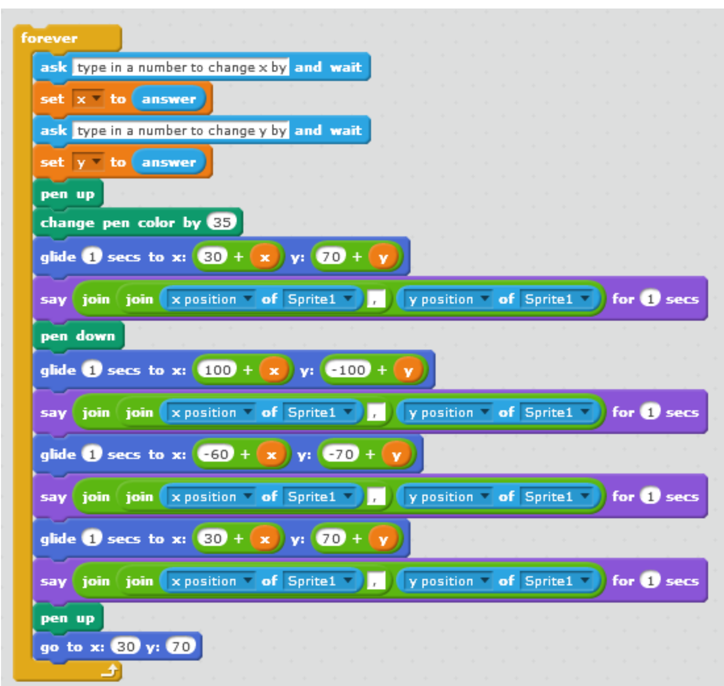


"glide", "say", and" join" are new parts to the code.

The first "glide" statement allows the pen to take 1 second to move to the fixed point (30, 70). Notice that the pen is up. For the other glide statements, the pen is down. It will take 1 second to draw each side of the triangle.

"say" will make a speech balloon or cartoon bubble quote.

"join" combines exactly two speech elements together. In the first statement with join, the inside "join" combines "30" with ",". The outside "join" combines "30," with "70". It would take 2 more "joins" in the statement to add parenthesis for the ordered pair.



"ask and wait" and "set to" are also new and used in the part of the program to translate the triangle.

The first "ask" requests the user to type in the amount of x translation using a bubble balloon, then waits until the user responds.

The set command takes the user's response and makes it equal to the variable "x". Not all variables have to be used for numbers.

Lesson 6 – Rotations

Name____________________________ Date_______ Block__

Run the Scratch programs:

"Quad 1-2-3-4EduardoO" https://scratch.mit.edu/projects/20523877/#editor

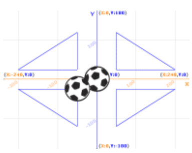
"Simple Triangle Translator" https://scratch.mit.edu/projects/69005220/#editor

"Polygon Rotation remix" https://scratch.mit.edu/projects/69009032/#editor

"Transforming with a Matrix" https://scratch.mit.edu/projects/1074767/#editor

Look at the new coding for the programs. Only 2 have rotations. What features do you like and dislike for each?

Quad 1-2-3-4EduardoO

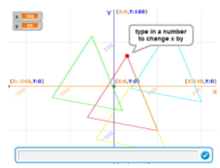


New Code

Likes

Dislikes

Simple Triangle Translator

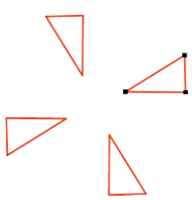


New Code

Likes

Dislikes

Polygon Rotation remix

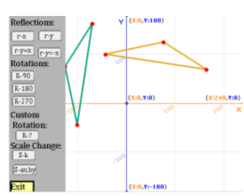


New Code

Likes

Dislikes

Transforming with a Matrix



New Code

Likes

Dislikes

What is the point (-5, 2) rotated 90 degrees about the origin?

What is the point (*x*, *y*) rotated 90 degrees about the origin? 180 degrees? 270 degrees?

Using trig and the functions sin and cos, what is the point (-5, 4) rotated 24 degrees about the origin?

Evaluation of the project will be accomplished using four instruments.

Pre and Post Survey on Computer Science Perspectives

- Determine student interests, feelings, enjoyment, and future plans involving STEM and computing.
- Check for perceptions changes by gender and race.

Daily Exit Ticket (Ticket Out the Door)

- Determine how well students learn or enjoy each lesson.
- Determine which is better, computing to teach geometry, or geometry to teach computing.

Common Assessments (copyright did not allow the printing of assessment problems).

- Students in regular geometry and geometry with computing take common problems on unit summative assessment.
- Results can be compared for students with and without computing instruction.
- Does coding help students learn geometry concepts?

Tracking Students’ Future Courses

- Data includes gender, race, AP Computer Science courses taken, and college major.
- A test for independence can be conducted for geometry with computer programming lessons and AP Computer Science.

Pre and Post Survey on Computer Programming Perceptions

Grade 6 7 8 9 10 11 12

Gender Female Male

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I have done some computer programming.					
I am good at computer programming.					
Some computer programming is fun.					
Computer programming can help me understand mathematics.					
I plan on taking a course in the future in computer science.					
I plan on taking AP Computer Science Principles in high school.					
I plan on taking a computer science course in college.					
I can see myself in a career where some computer programming is involved.					
I plan on majoring in computer programming or a computer science field in college.					
I can see myself working in a STEM career.					

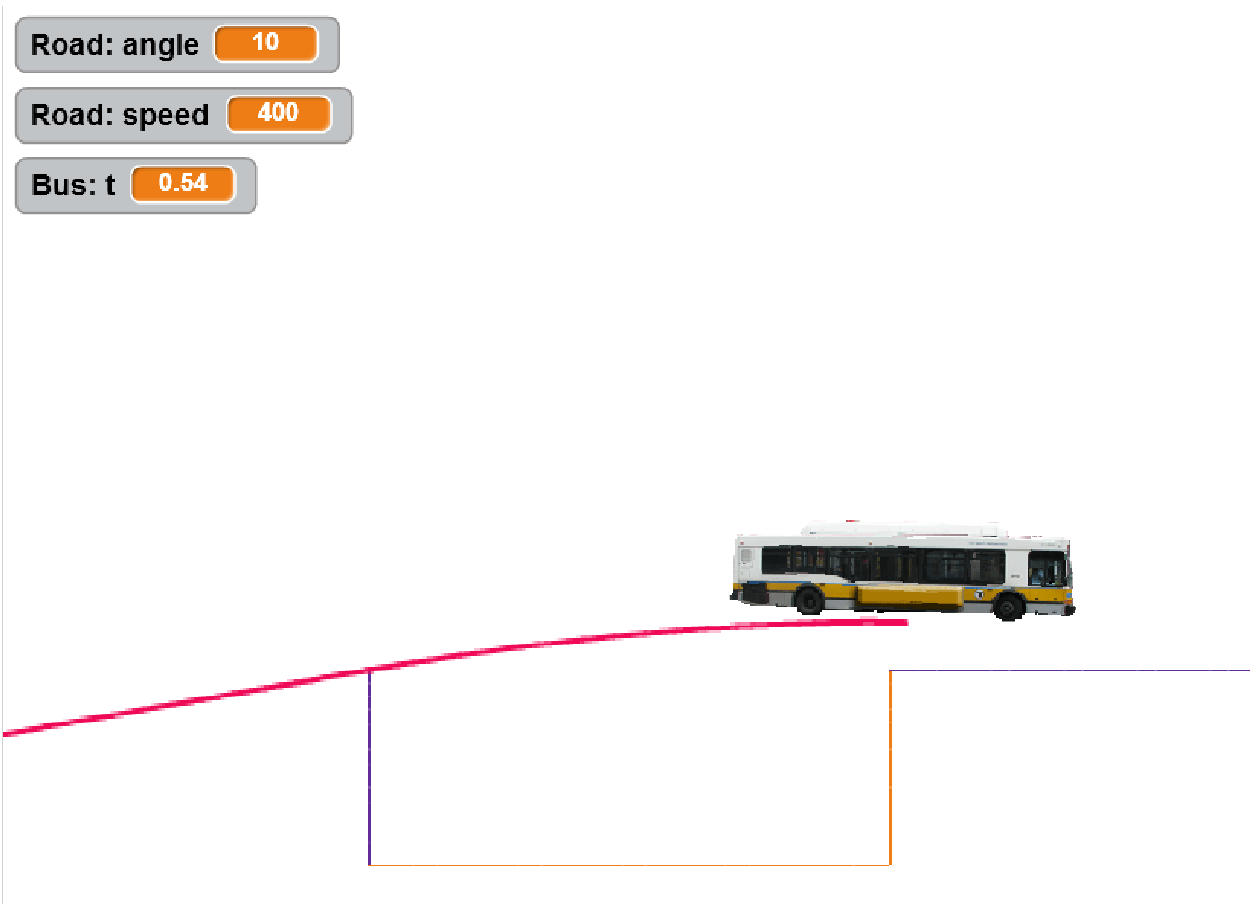
Daily Exit Ticket (Ticket Out the Door) Survey

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
This lesson made learning computer programming more enjoyable.					
This lesson made learning about transformations more enjoyable.					
This lesson helped me in learning computer programming.					
This lesson helped me in learning transformations.					

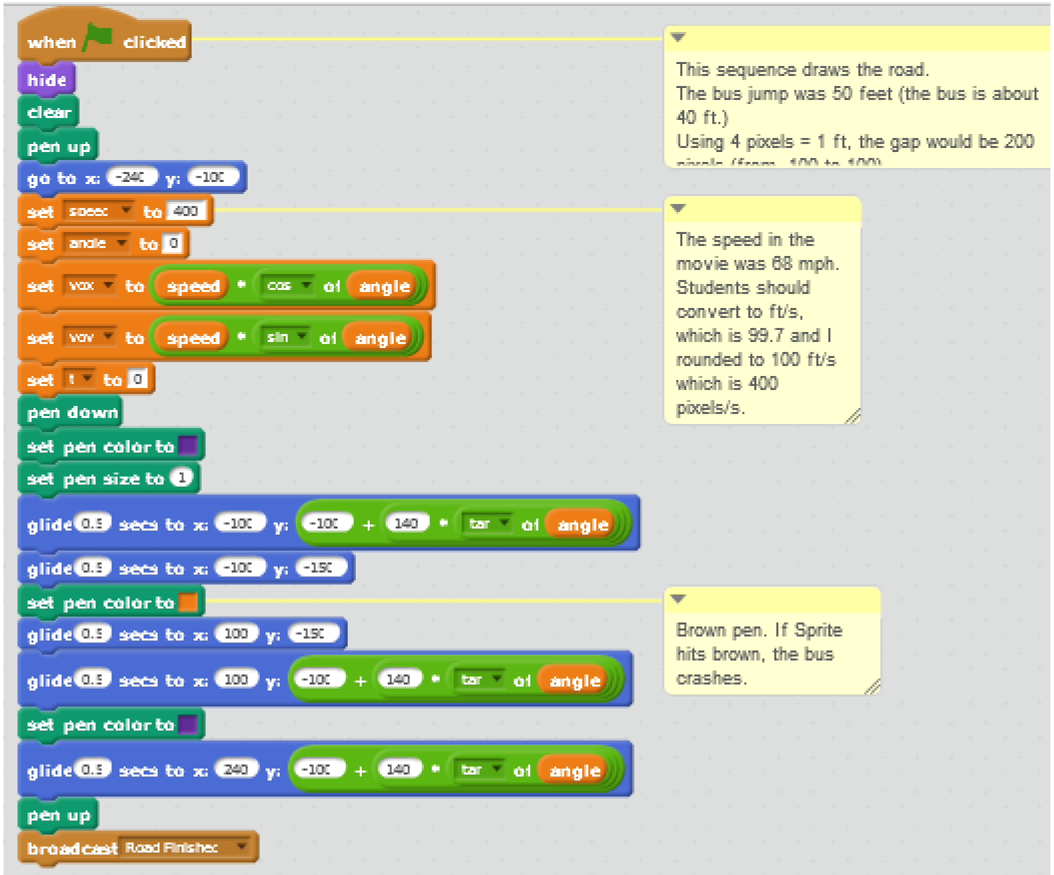
Trigonometry is now in the geometry curriculum. A PBL project launch for that unit is showing a two-minute movie clip from the 1994 movie “Speed” starring Keanu Reeve, Dennis Hopper, and Sandra Bullock. The scene is when the bus is approaching a 50-foot gap in the road. The hopes is that there is enough of a ramp there to jump the gap. The project: How can we as game designers create a simulation to determine if the bus jump in the movie is possible.

There are many ways to tackle the problem. The following Scratch solution has the ramp angle at 23 degrees (the students should measure this) and bus speed of 68 miles per hour. Using 4 pixels per foot, students must convert the distances to pixels, vertical and horizontal velocity to pixels per second and acceleration to pixels/s/s. The parabolic motion could also be determined using change in velocity. This program uses one sprite to build the road and one for the bus, which is from the Scratch sprite file. The angle and speed can be adjusted in the program. This motion is for a 10 degree angle and a speed of 68 miles per hour (400 pixels per second).

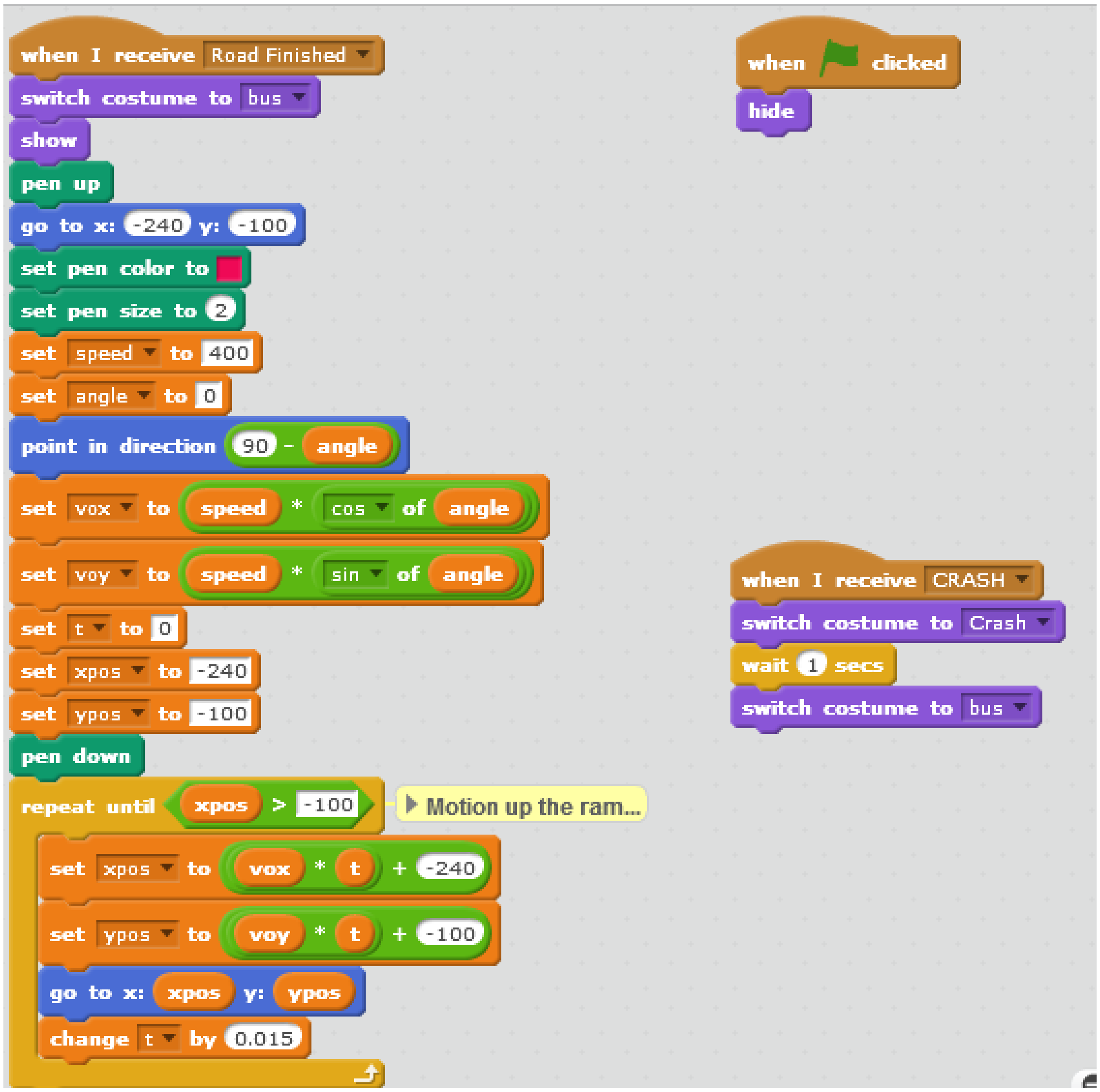
There are many different ways to solve this problem. There are also many ways to extend this problem in physics. This would probably not be used in 8th grade math.

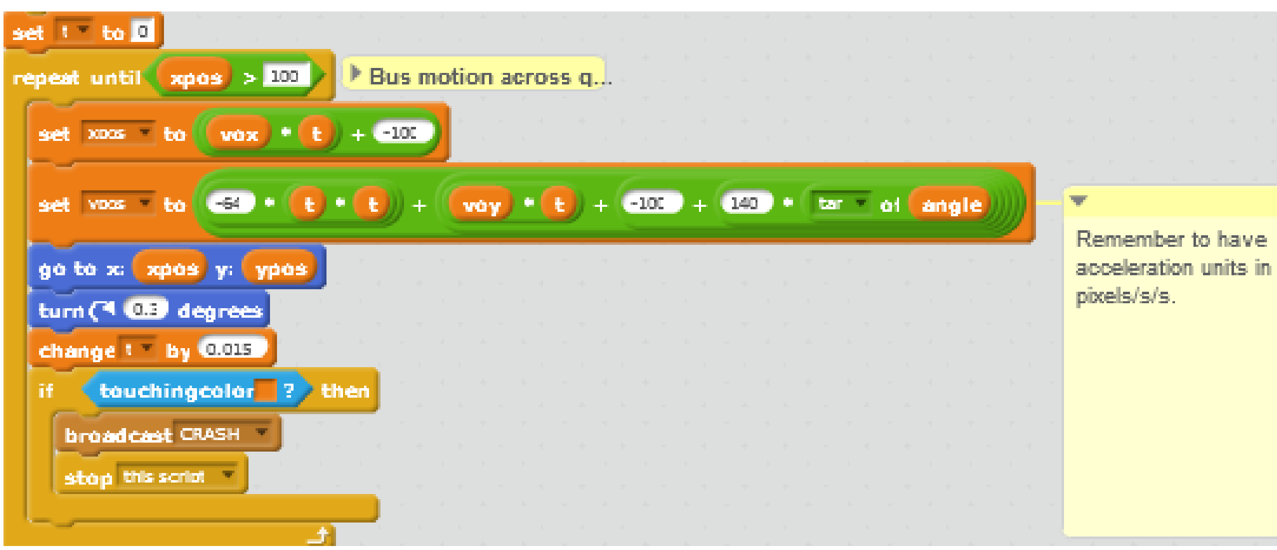


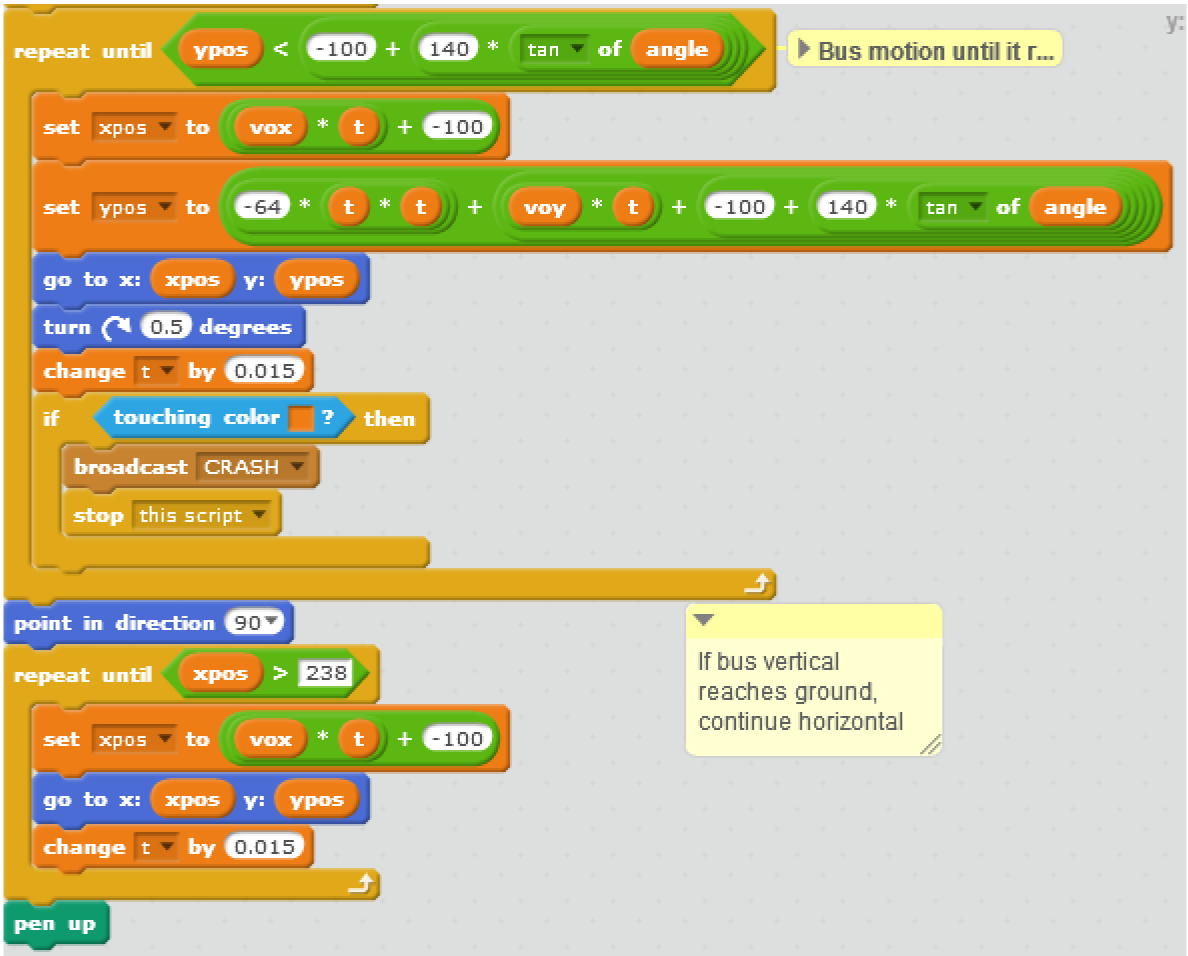
Script for building the road.



Script for motion of the bus.


when I receive Road Finished
switch costume to bus
show
pen up
go to x: -240 y: -100
set pen color to
set pen size to 2
set speed to 400
set angle to 0
point in direction 90 - angle
set vox to speed * cos of angle
set voy to speed * sin of angle
set t to 0
set xpos to -240
set ypos to -100
pen down
repeat until xpos > -100
Motion up the ram...
set xpos to vox * t + -240
set ypos to voy * t + -100
go to x: xpos y: ypos
change t by 0.015
when clicked
hide
when I receive CRASH
switch costume to Crash
wait 1 secs
switch costume to bus


set t to 0
repeat until xpos > 100
Bus motion across q...
set xpos to vox * t + -100
set ypos to -64 * t * t + voy * t + -100 + 140 * tan of angle
go to x: xpos y: ypos
turn 0.3 degrees
change t by 0.015
if touching color ? then
broadcast CRASH
stop this script
Remember to have acceleration units in pixels/s/s.


repeat until ypos < -100 + 140 * tan of angle
Bus motion until it r...
set xpos to vox * t + -100
set ypos to -64 * t * t + voy * t + -100 + 140 * tan of angle
go to x: xpos y: ypos
turn 0.5 degrees
change t by 0.015
if touching color ? then
broadcast CRASH
stop this script
point in direction 90
repeat until xpos > 238
set xpos to vox * t + -100
go to x: xpos y: ypos
change t by 0.015
pen up
If bus vertical reaches ground, continue horizontal
y:

Recommendations

Though the lessons can be used in either 8th grade or high school geometry, it is the recommendation of the author that they be used as a basis for an NSF STEM+C grant for high school geometry. Geometry teachers that teach AP CS Principles could build a large AP Program, similar to Algebra 2 teachers that teach AP Physics. There is the potential that gender and racial gaps could be greatly diminished while building large AP Computer Science programs. NSF, in its Strategic Goals for 2014-15, states the notion of supporting “high risk, high potential payoff” research. Substantially increasing the number of AP CS students (Hopefully leading to increased CS degrees), coding education for **all** students, and decreasing the gender and racial gaps would meet much of the driving force behind the grant. This is low risk, high potential payoff. **NSF’s broader impact criteria lists: full participation of women, persons with disabilities, and underrepresented minorities in science, technology, engineering, and mathematics (STEM); improved STEM education and educator development at any level…**

Potential Research Questions and Measuring Instruments

- <u>Pre/Post Survey</u> – Does implementing lessons utilizing computer programming in STEM classes encourage students to better enjoy programming and become interested in computer science fields?
- <u>Exit Ticket Surveys</u> - What formal and informal learning strategies and models are most effective in improving student engagement in STEM courses and acquisition of computing skills? (Is it better to use math to teach programming or programming to teach math?)
- <u>Common Assessments</u> - To what extent does implementing computer programming lessons in STEM classes increase student performance in STEM+C fields?
- <u>Tracking Students’ Future Courses</u> – Do lessons utilizing computer programming implemented in high school geometry increase the number and diversity of students taking AP Computer Science in Title 1 schools?

Steps that need to be continued are:

- Developing more lesson plans for HS Geometry (and 8th grade math and science).
- Making plans that are adaptable for Project Based Learning or traditional delivery.
- Conducting the proposed research to show benefits to schools and districts.
- Developing 4-day training for high school geometry teachers (and teams of 8th grade math and science teachers).
- Offering incentives for teachers at training, such as college or recertification credit, mobile devices, and the Scratch programming book.

Advantages of high school geometry computing lessons for grant proposal

- Transformations are part of national and state standards.
- Every high school has geometry.
- Almost every student takes geometry in high school.
- Not self-selecting (diverse student population).
- Computing is fresh in students' minds during registration for next school year.
- If geometry teacher also teaches CS course, easy to build program.
- Research on future AP CS enrollment is quick (months instead of 3 years).
- Intervention will lead to quicker upturn in college CS majors (3 years or more sooner).

Conclusion

There is a serious shortage of graduates with degrees in computing fields to fill the number of job openings for the next several years. Students who take AP Computer Science courses in high school are eight times more likely to major in computer science. NSF and the College Board are have initiatives to increase the number of students taking AP CS. The CS10K initiative seeks to train 10,000 computer science teachers for 10,000 schools. AP CS Principles is a new course that many students will hopefully find more inviting to begin a possible study in computer science. These initiatives will only place teachers in about 40% of the schools. Minorities are disadvantaged students will often still be left without options for taking computing courses in high school. Having computing content taught in geometry classes, which almost all students take in high school, are a promising way to give access to a more students and a much more diversified collection of students.

Resources:

1. National Center for Women and Information Technology. Retrieved July 26, 2015 from https://www.ncwit.org/sites/default/files/file_type/usnatgraphic2018projections.pdf

2. Zweben, S. & Bizot, B. (2015). 2014 Taulbee Survey: Relentless Growth in Undergraduate CS Enrollment; Doctoral Degree Production Remains Strong, But No New Record. *Computing Research* News, May 2015, Vol. 27/ No. 5. P. 4, 9, 14 http://cra.org/crn/wp-content/uploads/sites/7/2015/06/2014-Taulbee-Survey.pdf

3. National Center for Education Statistics. Digest of Education Statistics. Table 318.30. Bachelor's, master's, and doctor's degrees conferred by postsecondary institutions, by sex of student and discipline division: 2012-13. Retrieved July 28, 2015 from https://nces.ed.gov/programs/digest/d14/tables/dt14_318.30.asp

4. Timanovsky, Y (2015). ACM Education Council Task Force Tackles Issues of Diversity. Edubits – Digest of ACM Educational Activities. Association of Computing Machinery. ACM Inroads, June, 2015, Vol. 6, No. 2, P. 6 http://delivery.acm.org/10.1145/2760000/2756557/p6-timanovsky.pdf?ip=72.178.155.194&id=2756557&acc=OPEN&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E6D218144511F3437&CFID=697753923&CFTOKEN=67743070&__acm__=1438222612_4a94b37a73fa69e1743fcd5b4ea75fb7

5. College Board. AP Data – Archived Data 2013, National Report

http://research.collegeboard.org/programs/ap/data/archived/2013

6. Ericson, B. Detailed AP CS 2013 results: unfortunately, much the same https://computinged.wordpress.com/2014/01/01/detailed-ap-cs-2013-results-unfortunately-much-the-same/

7. Cuny, J (2012). Transforming high school computing: a call to action. ACM Inroads, June 2012, Volume 3, No. 2, P. 33, 34

8. Heitin, L. (2014). No girls, blacks, or Hispanics take AP computer science exam in some states. Education Week, January 10, 2014 http://blogs.edweek.org/edweek/curriculum/2014/01/girls_african_americans_and_hi.html?cmp=SOC-SHR-TW

9. Marji, M. (2014). *Learn to Program with Scratch – A Visual Introduction to Programming with Games, Art, Science, and Math*, No Starch Press, San Francisco, CA

10. NSF, Strategic Plan for 2014-2018, http://www.nsf.gov/pubs/2014/nsf14043/nsf14043.pdf

Acknowledgement

A huge thank-you to Dr. Truell Hyde, Jorge Carmona Reyes, Dr. Li Wang, Dr. Lorin Matthews, and Lisa Tarman.
This work was supported by NSF grant No. 1262031

About the Author

Gary R. Stark received his B.A. degree from Winona State University, MN in 1983, majoring in mathematics and physical science. Mr. Stark received the M.A. degree from the University of Northern Colorado in mathematics in 1997, and is two courses shy of a second M.A. in physics from the University of Virginia.
He began his teaching career in 1983 at Lac Courtes Oreilles – Ojibwa Schools in northern Wisconsin, teaching math, science, and computer programming. He moved to Estes Park, CO in 1985, where he taught at Estes Park and Longmont High Schools. From 2001 to 2003 he directed a GEAR UP grant at Adams City High School in Commerce City, CO where he also was the math coach and initiated the AVID program in the district. From 2003 to 2013 he taught mathematics and science at Estes Park High School. He is currently teaching mathematics at Greeley Central High School in Greeley, CO.
Mr. Stark has participated in the Research Experience for Teachers (RET) programs at Cornell University, NY and Baylor University, TX. He has also participated in the Academy Creating Teacher Scientists (ACTS) program at Princeton University, NJ. He has been a member of the National Council of Teachers of Mathematics, National Council of Supervisors of Mathematics, National Science Teachers Association, American Association of Physics Teachers, and the National Education Association.