Project ECSITE Activity: Math Animation Challenge

Subject Areas: Algebra 2, animation, programming

Associated Unit: Robotics and Algebra 2

Time Required: 60 minutes

Group Size: <30 students

Expendable Cost per Group: US$ 0

Learning Objectives: Student become more familiar with programming (in javascript), have to fit a line to several goal points, animate a character and make a video game-like program.

Computational Thinking Connection: Computer graphics, animation, programming

Materials List:

Per Student:
- Netbooks for every student

Introduction/Motivation: Building off the desmos computer graphics lesson, this lesson goes a step further and introduces real programming, animation and shows students how to apply a polynomial function to create a character trajectory.

Vocabulary Definitions:

polynomial function - an expression of finite length constructed from variables (also called indeterminates) and constants, using only the operations of addition, subtraction, multiplication, and non-negative integer exponents

Procedure: this is the outline I used for the lesson:

Outline of Presentation

- Review Desmos art project
  - Now: animation
- What is Khan Academy
  - You can do it too
  - Javascript | Canvas
  - Be creative, explore
  - Code executed in real time
5 Minute Intro to Programming
- // Green = comments. how to uncomment
- var = variable
- multiplication must be specified with *, and power with pow(a,b)
- var drawObstacle = function()  FUNCTIONS
- var draw = function()  ANIMATION LOOP
- more info: documentation at bottom of webpage

Animation
- Series of images flashing on screen fast
- Clear screen each frame
- FPS

The challenge:
- Have the moving dude hit all 4 obstacles with one trajectory
- How?
  - Polynomial functions
  - Remember to consider end behavior: up-up, down-up, etc.
  - Use this form:
    - y = a(x-b)(x-c)(x-d)(x-e) + f
      - where a vertically stretches the function
      - b through e are the x-intercepts
      - f moves the function vertically
  - My solution:
    - y = 0.006*(x+9)*(x+3)*(x-1.5)*(x-9)+3;

You try it!
- Go to http://khanacademy.org/cs/new
- Do this simple code:
  - // this is a comment
  - fill(255, 0, 0);
  - rect(66,123,50,50);
- Now go to http://goo.gl/5ktck ← the template
  - or http://www.khanacademy.org/cs/math-animation-challenge/1113135891
- Scroll down all the code and get a quick sense of it. stop at bottom
- At bottom - Section 10
  - Uncomment drawDude
  - Uncomment drawObstacles
- Scroll up to Section 9
  - Find the part that says “Add your formula here”
• Change the equation from $y = 2x$ to a 4th degree polynomial function, of the form $y = a(x-b)(x-c)(x-d)(x-e) + f$
• Use desmos.com if you need to
• Tweak until all 4 parts are touched

• Now have some fun...
• Make it pretty
  • Go back to section 10
  • Comment out drawGrid
  • Uncomment drawBackground

• Change images
  • Go to section 4
  • Click on the text inside “getImage” and hover over the popup square
  • Choose new images to your liking

• Change speed
  • Go to section 1
  • Lower the dude_speed variable
  • Now go to section 9 and add this formula under the one you typed in (the part that says “Add your formula here”):
    • $dude\_speed += 0.001 * y$;
  • Do whatever programming/hacking you want to in the time remaining

Safety Issues: None

Attachments: Handout.pdf

Adaptations and Extensions: n/a

Assessment:

In this 1.5 hour lesson I taught students the very basics of programming in Java script, I discussed how animation worked and then gave them a template the replicated a graphing calculator or the website desmos.com, which they are now very used to. I then allowed them to make some small changes to the code so that they could turn on parts of my programming – this gave them the feeling they were doing a lot. I then had them change the math function of the plotted line into a polynomial function that would allow the animated dude to float over all 4 obstacles. When he reached an obstacle, the object would shake, dissapear or turn into something better. This gave a game-like quality to the activity that the kids really like. I then allowed them to customize the code and hack at it, which many really took a liking too.
If I could change something I would actually have provided more things for them to do to the code. I actually made it too easy for a majority of the class. On the other hand, there were a few students who found it tough and it was just the right amount. The smartest kids finished super fast but where motivated enough to hack at it by themselves.

At the end we shared everyone's programming creations on the projector and it went super well. I'm very happy with this activity and very impressed with the power of processing.js – what Khan Academy uses for programming live!

**References Used:**


**Contributors:**
Dave Coleman

**Supporting Program(s):**

NSF and ECSITE