Art + Code

*parts and crafts * Fabville• spring 2018*

Contact: William Macfarlane

Parts and Crafts • Center for Semiconducted Learning
will@partsandcrafts.org • 617-207-8016 • Makerspace • Unschool • Workshop • Camp • www.partsandcrafts.org

Workshop description
A free six-session introduction to the basic concepts of computer programming using Javascript and the p5.js programming environment with a focus on code as a tool for visual arts and creative expression.

Javascript is a flexible and powerful programming language that can run in every web-browser. These features make it a popular tool for developing software of all kinds as well as a convenient and accessible environment for learning how to program.

Target Audience
Adults and high-school age students who are interested in the creative and expressive potential of computer programming.

No particular technical skills beyond basic computer literacy are expected for this course.

Course objectives
To understand coding as a form of structured problem solving and to be able to use computational tools, numeric processes and data to produce creative works of visual and interactive art.

To provide a low-stakes introduction to coding for people considering further study and the pursuit of a software career.

Materials Needed
These projects can all be done using the p5.js web editor alongside other freely available open-source software tools and data.

Students will need a relatively modern computer with an internet connection and a web-browser. Most computers manufactured in the last 10 years will suffice.

Software Setup
Students will use the p5.js web editor as their primary development tool and should create an account on the p5.js website to save their work.
https://alpha.editor.p5js.org/

We will also use github for sharing code and resources and hosting projects. Students should create a github account before the beginning of the clas.
https://github.com/

Week 1
What is code?

- Using a text-editor
- Writing/running code
- Javascript syntax
- Color and pixels

Week 2
Animating with Numbers

- Variables and arithmetic
- Conditionals
- Animation and motion

Week 3
Logic and Flow

- Loops and iteration
- Variables and conditionals
- Events

Week 4
Putting pieces together

- Functions
- Program composition
- Events and Interactivity

Week 5
Structured Data: Lists

- Lists
- Arrays
- Loops
- Iteration

Week 6
Stuctured Data: Objects

- Objects
- Functions
- Arguments
- Data sets

Week 1

What is code?

- Using a text editor
- Writing/running code
- Javascript syntax
- Color and pixels

Questions: What is code? What is javascript? How do we write programs and what does it mean for a program to be run?

Activities: List the layers of software from cpu to javascript interpreter. Use the 'view source' tool to look at code examples in the wild and us the p5.js web editor to make our first programs.

Key Terms: INTERPRETER: a computer program that directly executes, i.e. performs, instructions written in a programming or scripting language • CANVAS ELEMENT: an html container for graphics usually drawn by Javascript • PIXEL: the smallest controllable element on a computer display • CARTESIAN COORDINATES: a coordinate system that specifies the location of a point in terms of its distance in two dimensions from a pair of perpendicular lines • RGB COLOR MODEL: an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors

Projects: Using the p5.js web editor to write Javascript, draw basic shapes on a canvas element. Learn the basic shape primitives: line, ellipse, rectangle, triangle. Learn the basic RGB/A color model and use these basic commands to draw simple shapes and designs including a smiley face and a creature.

Challenges: Use the shape primitives to draw two frames of an animation. Perhaps your smiley face might open its mouth or close its eyes, or your creature might walk around. Write code for each of these frames and switch back and forth between them by commenting sections of your program out.

References:
https://p5js.org/reference/#/p5/background
https://p5js.org/reference/#/p5/fill
https://p5js.org/reference/#/p5/stroke

https://p5js.org/reference/#/p5/ellipse
https://p5js.org/reference/#/p5/line
https://p5js.org/reference/#/p5/rect
https://p5js.org/reference/#/p5/triangle

https://www.youtube.com/watch?v=8j0UDiN7my4

Week 2

Animating with Numbers

- Variables and arithmetic
- Conditionals
- Animation and motion

Questions: How can we give our graphics motion and life? How are position and motion represented to the computer? How do we store and change the state of our program?

Activities: Look at standard frame-rates for various media. Review useful properties of arithmetic. Present the modulus operator. Discuss data types and how javascript is different from other programming languages.

Key Terms: VARIABLE: a value that can change, depending on conditions or on information passed to the program • VARIABLE SCOPE: the region of a computer program where a variable name is associated with a value is valid • BOOLEAN ALGEBRA: the branch of algebra in which the values of the variables are the values true and false • VECTOR: a quantity having direction as well as magnitude

Review: Basic program syntax. Make a new project, draw a set of shapes to meet a given specification of shape, color, size, and arrangement.

Projects: Coding motion. Use the frameCount variable and the modulus operator to time a two-frame animation and use boolean variables and conditionals to switch between frames. Use number-type variables to keep track of position and speed to program a bouncing ball animation.

Challenges: Use a numeric variable to cycle through a multi-frame animation. Write code so that your bouncing ball changes color randomly when it bounces off the wall. Add an additional bouncing ball. Use a random variable to make a "dice" program that displays a 1 to 6 circles on the screen whenever it is run.

References:
https://p5js.org/examples/control-conditionals-1.html
https://p5js.org/examples/data-variables.html
https://p5js.org/examples/data-variable-scope.html

https://processing.org/reference/modulo.html
https://p5js.org/reference/#/p5/frameCount
https://p5js.org/reference/#/p5/random

https://www.youtube.com/watch?v=LO3Awjn_gyU

Week 3

Logic and Flow

- Loops + Iteration
- Variables and Conditionals
- Events and Functions

Questions: What path does the interpreter take through the code that we write when we run it? How can we write code that changes this logical flow?

Activities: Introduction to Boolean logic and truth tables. Disect the syntax of javascript if-statements and p5.js functions and blocks. Pythagorean theorem as distance formula.

Key Terms: LOOP: a sequence of instruction s that is continually repeated until a certain condition is reached • LOOP INDEX: a variable tracking how many times a loop has been iterated • EASING: altering the rate of change of a quantity to make animation less severe and more lifelike • unique values with unexplained meanings that should be replaced with named variables

Review: Definition of a vector. Rewrite bouncing ball program to use rectangles instead of circles. Use a variable to control rectangle width and height and write the bounce code so that it pays attention to this variable and works correctly for all reasonable values. Clean up our code replacing all magic numbers with variables.

Projects: Fill the screen with random shapes every frame. Create “sparkles” effect with random ellipses following the mouse-pointer. Draw a checkerboard pattern on the screen using nested loops. Make simple easing procedure to cause a shape to follow the mouse pointer around. Use mousePressed() event function to restart the program.

Challenges: Make a checkerboard with randomly varying overlapping squares. Draw a circle on the screen that changes color when your mouse is overtop of it and changes the background color when you click on it.

References:
http://easings.net/
https://en.wikipedia.org/wiki/Pythagorean_theorem

https://p5js.org/examples/structure-loop.html
https://p5js.org/examples/control-iteration.html

https://p5js.org/reference/#/p5/dist
https://p5js.org/reference/#/p5.Element/mousePressed

Week 4

Structured Data: Lists

- Lists + Arrays
- Loops
- Iteration

Questions: What are loops and repetition good for? How can we define data in our programs to take advantage of iteration?

Activities: Imagine adding a third bouncing ball to the bouncing ball program. A fourth, fifth, sixth. Identify the steps of this process and the common elements that we have to add for each one.

Key Terms: ARRAY: a data structure consisting of a collection of elements (values or variables), each identified by their position in the collection • STATE: the collection of variables and data structures that define a program’s behavior • INDEX: a number that maps a value in an array to its position in the array which can be used to access that value

Review: Loop syntax. Use console.log() function to test simple loop behaviors. Create random-walker ellipse and turn off background() to draw trails. Add “alpha” to color-model and review differences between HSV and RGB color-space.

Projects: Rewrite the bouncing ball program to use arrays of values to store the data for an arbitrarily large number of balls. Do the same for the random walker. Make sure all properties and property-ranges are stored in variables that can be easily modified by the user. Write a program that fills the screen up with circles which will disappear when the user clicks on them.

Challenges: Add a time-step feature to the random walker program to make it draw images faster. Modify the way colors are chosen so that the color of the walker trails changes gradually over the life of the program. Add alpha to the trails and experiment with different properties and values to generate different kinds of shapes. Write a program that generates a randomly moving ball every time the mouse is clicked.

References:
https://p5js.org/reference/#/p5/print
https://p5js.org/examples/arrays-array.html

https://www.youtube.com/watch?v=VIQoUghHSxU
https://www.youtube.com/watch?v=RXWO3mFuW-I

https://en.wikipedia.org/wiki/Random_walk

Week 5

Functions and Composition

- Functions
- Arguments
- Program organization
- Refactoring

Questions: How are the commands that we use in our programs originally written? How do we define our own commands and operations? As our programs grow in complexity how can we keep them organized and understandable?

Activities: DRY Principal: look through our existing code to find places where we repeat ourselves. Look at built-in p5.js functions and map their syntax. Contrast mathematical definition of function with programming definition.

Key Terms: DRY PRINCIPAL: "Don't Repeat Yourself," a software engineering principle designed to eliminate redundancy and formalized as "every piece of knowledge must have a single, unambiguous, authoritative representation within a system" • FUNCTION: a named section of a program that performs a specific task and can be referenced ("called") elsewhere in the program • ARGUMENT: also called "parameter", a special kind of variable, used in a function to refer to one of the pieces of data provided as input to the function.

Review: Write a program that fills the screen with random sized boxes that randomly jitter while the mouse is hovering over them. Learn the keyPressed() function and write a simple drawing program where lines are drawn when the mouse is pressed and keys on the keyboard can be used to switch colors or to clear the screen.

Projects: Simplify existing code by creating functions to define repeated operations, including bounce() and changeToRandomColor() for bouncing ball. Break bouncing ball and random walker programs down into a series of well-labelled functions that take an array index as an argument and perform the labelled operation [move(), bounce(), changeColor(), etc] on the item defined by the array index.

Challenges: Add friction and forces to bouncing ball program so that the balls slow down over time and gain new random speeds whenever the space key is pressed. Generate a random selection of balls when the mouse is held down and modify the properties of the balls over time to create an interesting particle effect.

References:
https://p5js.org/examples/structure-functions.html
https://p5js.org/reference/#/p5/keyPressed

Week 6

Structured Data: Objects

- Objects
- Properties + Methods
- Types and Arguments
- Further Exploration

Questions: What are objects? What is object-oriented programming? How can we organize our data in our code to more closely resemble the way we think about it?

Activities: Look at our existing codebase for our bouncing-ball/particle systems and our random walkers -- identify the objects in the program and the properties that these objects can have and the things these objects can do and have done to them.

Key Terms: OBJECT: a form of structured data that defines an item's state in terms of property variables and behavior in terms of functions • PARTICLE SYSTEM: a technique in modelling and computer graphics that uses a large number of very small objects to simulate certain kinds of "fuzzy" phenomena and natural processes • CLASS: a template for creating objects with predefined properties and methods • INSTANCE: a concrete occurrence of any object, existing usually during the runtime of a computer program

Review: Definitions of vector, force, acceleration, speed, and position. Newton's Laws for animation and simulation. Write a program that generates a set of bouncing balls with random speed and direction and draws lines between all of the balls. Write code so that you can toggle the visibility of the balls and the lines with the keyboard.

Projects: Rewrite bouncing ball code using "particle" objects with speed, position, color and size properties. Create a particleSystem array to store these objects. Write functions for "move", "bounce", "friction", "shrink", "grow", "randomWalk" and others that take particle objects as arguments. Create "fireworks display" particle system.

References:
https://p5js.org/examples/objects-objects.html
https://p5js.org/examples/objects-array-of-objects.html

https://p5js.org/examples/simulate-particle-system.html

Further Exploration:
p5.js website, examples, library, and community.
https://p5js.org/
Daniel Shiffman/The Coding Train
https://www.youtube.com/channel/UCvjgXvBlbQiydffZU7m1_aw
General Q/A:
https://stackoverflow.com/