

What are we doing here?

This is designed to build on the understanding you developed in A06, in order to help you produce A07, in the run-up to the final project.

What does this in-class activity demonstrate?

- Your ability to solve problems with code.
- Your ability to translate a set of instructions from natural language into a programming language.
- Your ability to program in p5.js, including:
 - Creating a canvas and a background.
 - Working with variables.
 - Working with conditionals.
 - Using common functions and system variables for p5.js.
 - Incorporating input (random, frameRate, mouse position, mouse button, key press, key value)
- Your ability to produce original animated compositions
- Your ability to consult outside resources (documentation, textbooks, web searches) to solve programming problems
- Your ability to coordinate with your peers to solve programming problems

Am I allowed to look in the p5.js text book? Can I search Google for the answer?

Absolutely, yes, but if you use someone's else code, use a comment `///
you found the code.`

Am I allowed to ask other people for help?

More than that, you'll be reliant on other people. Work near one another. When you get stuck, ask someone for help. If you see someone stuck, ask to help them.

After this exercise you should know:

How to generate a random number.

How to know if the mouse button is being pressed.

How to know what the mouse's position is in the current frame.

How to know how much the mouse moved between last frame and this frame.

How to know if a certain key is being pressed.

How to use how many frames have been displayed since the program started as a variable.

How to use conditionals (if/else and for).

Step Zero

1. Open OpenProcessing & login. Go to our class. Find the EXAMPLE sketch in Activity 07 (part 03).
2. Fork the sketch.
3. Fill in the relevant fields and save the sketch.

Step One: Setup

1. Declare a variable that will represent the color of the background. Set it to 0 initially.
2. Declare a variable that will represent the alpha value of the background. Set it to 255 initially.
3. Create a canvas that is the height and width of the window.
4. Set the background with the variables you established earlier.
5. Set the frame rate to 30 frames per second.

Step Two: Set the stage

1. Make the program draw the background each time draw() runs. Use the variables set earlier to set the background.
2. When the mouse is pressed, check to see the mouse's position. If it is in the top half of the canvas, have it decrease the background's color value by 10. If it is in the bottom half of the canvas, have it increase the background's color value by 10.
3. When there is a key press, check to see what key has been pressed. If it is the up arrow, then increase the background opacity. If it is the down arrow, then decrease the background opacity.

Step Three: Problems to Solve

- Use the random() function to place a randomly sized shape (with attributes of your choice) on the canvas at a random x,y location every 10 frames.
- Draw a number of shapes across the canvas at a certain interval pixels apart.
 - The number of shapes should be controlled by a variable that can change with each iteration of draw().
 - The shapes should somehow vary in color or style.
- Create an object that moves across the canvas. When it reaches the edge of the canvas, have it 'bounce' off the edge of the frame so that it doesn't go off the canvas.

Step Four: Bonus Round

- Revise the function that changes background color value based on mouse position. Change the interval it changes the mouse value – have it scale with the motion of the mouse across the canvas. If it moves a large distance between the last frame and the current frame, have it increment the value by a lot. If it moves a small distance, have it increment the value by a little.
 - Add a line that indicates last position and current position as a means of visualizing the change.
- When the spacebar is pressed, have it reverse all color values in the sketch. HINT: Consider the values as a number line, with the middle point being at 127. If the background color is 50, make it 204. ($127 - 50 = 77$. $77 + 127 = 204$.)
- Use the input from the mouse wheel as a control to increment one of the counters.
- Use the `dist()` function to activate a behavior or change in the sketch.