

Useful Resources:

<https://processing.org/tutorials/objects/>

Instructions:

All parts of lab questions that have **R** at the beginning are mandatory, and their submission by each student is required. The parts of each question that begin with **O** are open-ended enhancements for further investigation into course material. These should be completed only after the mandatory parts of the lab are completed, and are not explicitly required for submission. If you choose to complete further optional add-ons, you are welcome to explore your own options, and not just the suggestions provided. Submit a zip document to the dropbox folder L7 on LEARN. The document should contain separate folders for the following files:

- **L7_Turtle**
- **L7_Fountain**

Lecturette:

When to use classes (lots of the same thing on a canvas, varying slightly and independent of each other)

Creating a class; constructors, methods, format, calling a class method.

For this Lab, we won't worry about complex behavior like collisions; so don't use the Fisica library in either question.

1. Turtle

- a. **R** Load the provided SVG file named turtle.svg. Create, using this PShape, a class called Turtle, which defines a turtle as an instance of the loaded shape.

Include in your class definition parameters that represent an object's;

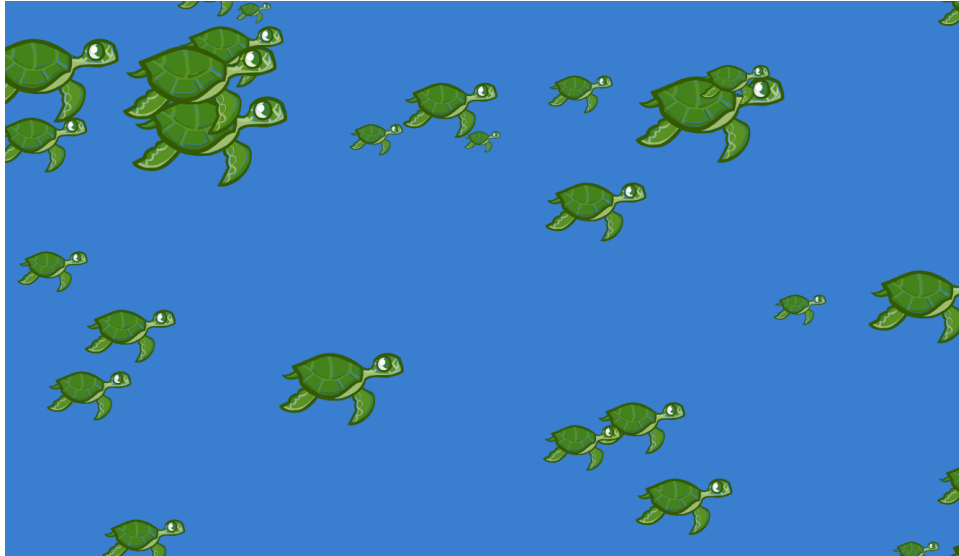
- Size (relative to the initial size of turtle.svg)
- X velocity
- Y velocity
- X position
- Y position

Initialize these by randomizing your constructor function within your class.

Define 2 methods within your Turtle class;

- One that will display the image of the instance, in the correct position
- One that will update the image position based on velocity

Using this definition you have just created, make an array of type Turtle so that a very large number of instances can be shown on the canvas easily at the same time, acting independently of each other. Do this by first defining an array of turtles, then looping through the entire array, drawing the turtle using the display method, and then updating the position using the second method you defined. Be sure to include the SVG file in the saved folder.



2. Fountain

The easiest way to implement numerous water droplets (the fountain you will implement should have at least 100) is using an array with each element as an instance of a class, just like in the first question. This will also allow you to ‘recycle’ the drops when their position is out of the bounds of the canvas.

- a. **R** Using the starter code provided, create a fountain that spurts water out of its spout by defining a Droplet class that represents the information about 1 water droplet. This class should have the same parameters (size, 2 positions, 2 velocities) and methods (to draw position and to update position) as the first question of this lab. Your Droplets should be initialized at the point (150, 275), the top of the spout. (This part is very similar to question 1).
- b. **R** To make your fountain more realistic, give the water droplets varying colours by adding a colour parameter to your class, and initializing the colour randomly.
- c. **R** Add gravity to your fountain, so that if water will act with projectile motion by spurting upwards out the top of the spout (at position 150, 275). This will act very similarly to when you created a bouncing ball without using Fisica, by increasing the velocity within your updating method of the Droplet class.
- d. **O Challenging** As you may have noticed, the fountain starts off spurting droplets in “waves”. Try to update your code to prevent this from happening.