

**Useful Resources:**

SVG files	<code>shape();</code>	<code>translate();</code>
PShape	<code>pushMatrix();</code>	<code>rotate();</code>
<code>loadShape();</code>	<code>popMatrix();</code>	

**Instructions:**

**Be sure to save any attached files (SVG files called within your function) in the same folder as your code.** 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 L0 on LEARN. The document should contain separate folders for the following file(s):

- **L5\_Elephant1**
- **L5\_Elephant2**
- **L5\_Robot**

---

**Lecturette:** Example on how to load translate shapes and rotate shapes using `pushMatrix()` and `popMatrix()`

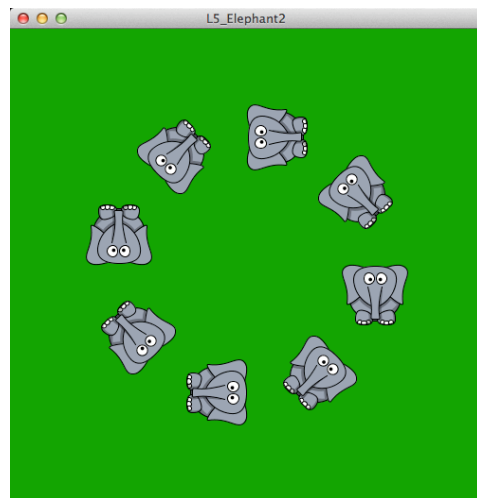
---

**1. Elephant1**

**R** Download the given SVG file called `Elephant.svg` and the starter code, which reads in `Elephant.svg` and draws an organized herd of elephants. Create a grid of elephants across the whole canvas, spaced evenly (as shown below). To draw each elephant, you must use the statement `shape(elephant, 0, 0)`. You cannot substitute other numbers for 0. You must then use the correct sequence of calls to `translate()`, `pushMatrix()` and `popMatrix()` to cause the elephants to appear in a grid. This can all be done within your `setup()` function, using nested loops.

**2. Elephant2**

**R** Using the same starter code given, create a circle of elephants around the middle of your canvas facing the next elephant within the circle (as shown below). Use `translate()` once at the beginning of your code (to move the middle of all subsequent translations the middle of your canvas), and then `pushMatrix()` and `popMatrix()` within a loop to individually translate the elephants around the circle and rotate them towards the next elephant. As in part 1, you must use the statement `shape(elephant, 0, 0)`, and cannot substitute other numbers for 0.



## 2. Robot

**R** Write a sketch that makes a robot arm with different sections that can move on “hinges”, joints that connect the portions. For this you may simply use lines for the arm sections. Each joint, when moved, should affect all subsequent joints of the body. For example, if the base joint were moved, all smaller joints of the arm would also be moved, but if the last joint is moved, none of the larger joints would be affected. Starter code has been provided, which uses GUI sliders to change the angles of the joints. By changing the values of the sliders, the angles should change. For the sake of simplicity, you may assume that the base of the robot arm is stationary and cannot be moved.

**HINT:** The starter code contains exactly one call to `pushMatrix()` and `popMatrix()`. You can complete the sketch without adding any additional calls to these functions. Instead, let `translate()` and `rotate()` do all the work, and do all your calculations within the same block. This way, you should not have to compute all the individual x and y positions for the individual parts.

