1 Lab 2

Goals

The goals for this lab are:

1. Practicing using Processing

2. Practice creating shapes in Processing

3. Understand draw order and command ordering

4. Understand the Processing 2D coordinate systems

5. Make a picture of a creature!

Modality

This is a pair-programming lab - please form teams of two people and trade off typing in commands and giving instructions to they-who-are-typing.

Details

Task: You must create an image using Processing of a creature. Your 'sketch' of the creature must:

- be created using at least 4 different shape types (see the list below).

- have at least 5 appendages. Examples of an appendage include: ears, legs, arms, wings, teeth, antennae, etc.

- be of size 400 by 400 pixels

- be in color.

In addition, try to make your creature look expressive (i.e. how does your creature feel at this moment of being drawn?)

First:! Please start by sketching your creature on paper (include coordinate labels to help you stay oriented). For example:
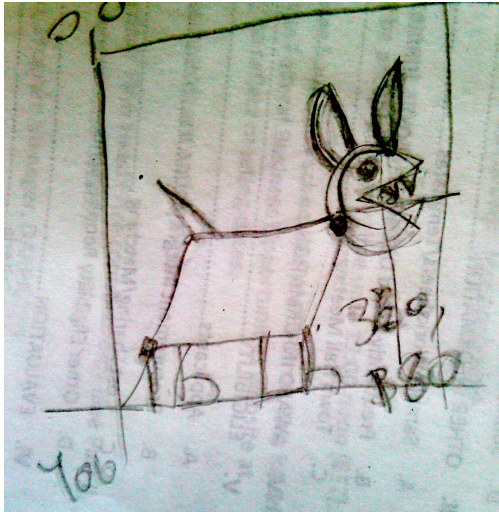

Figure 1: A sketch of an angry bunny.

Next: Working with your partner create a sketch in Processing of your creature. *Enjoy!*
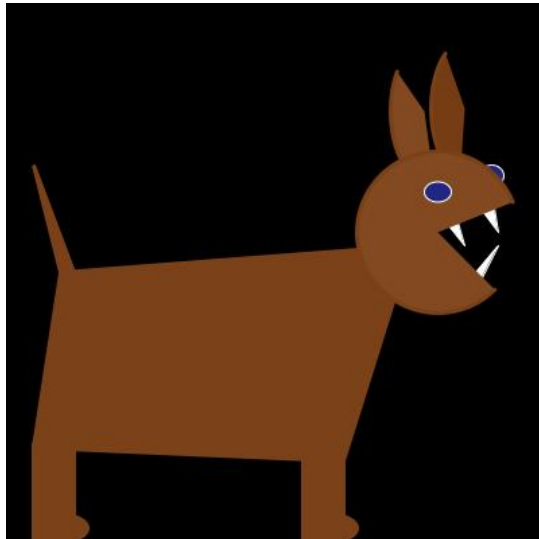

Figure 2: Example Creature created in Processing. It is an angry bunny.

Demo:

In order to receive credit for this lab, you and your partner must demo your sketch to your instructor along with handing in the image and sketch of your creature via handin. Ask your instructor for details.

2 Resources

Some of this information will help you on your lab.

To create shapes in Processing, you have various choices. Some of them are:

1. `line(x1, y1, x2, y2)`

2. `triangle(x1, y1, x2, y2, x3, y3)`

3. `quad(x1, y1, x2, y2, x3, y3, x4, y4)`

4. `rect(x, y, width, height)`

5. `ellipse(x, y, width, height)`

6. `arc(x, y, width, height, start, stop)`

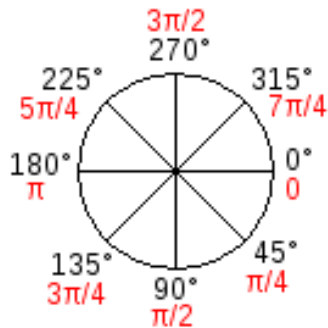**Note clockwise ordering! and remember about degrees and radians.**

Figure 3: Conversion of degrees to radians - and the 'rotation' convention in Processing.

Note that when drawing a shape you can control some of its properties. Commands that are useful are:

- `smooth();`

- `stroke(r, g, b);` or `stroke(grey);`

- `fill(r, g, b);` or `fill(grey);`