UNIT 23. USING PROCESSING IN JAVA

For the most part, you will end up leaving much of Processing behind as you move to pure Java. However, Processing is very useful in how much power it gives you over graphics and user input. As such, you may still want to use your Processing knowledge to add graphics to your programs.

To get Processing to work inside Java, you have a few problems to solve. The main problem is to import all that Processing capability into your Java system. Luckily, you can do this by adding a library. Then, you need to learn how to write that code into your program.

Unfortunately, how to link an external library to you project is highly specific to the actual IDE that you use. If you dig around, you should be able to find out how to link a JAR file – a Java ARchive.

In Eclipse, it is quite simple. Start a new project as per the code above. In the Package Explorer in the left-hand-side of Eclipse, right click on your project name (the top-level tree hierarchy); for example, `FirstProject`. Look for the item in the menu called Build Path – this tells Java which files to include in the project. Don't use the `import` command as this actually copies everything fully into your project, and simply won't work.

Click on Build Path and then click Add External Archives. A window should pop up that lets you select a file.

Navigate to your Processing folder where you installed it on your computer. Once there, go into the `core` subfolder and then `library`. Select the `core.jar` file and open, to add it to your project. You have just include the core Processing capability into your Java project!

You can test this by trying to `import` a Processing library into your project. Open the `FirstProgram` source file, and at the very top, add the following line:

```
import processing.core.PApplet;
```

Try to run your program. If you get an error (e.g., that the import cannot be resolved), then your Processing JAR file is not properly linked. Try again or ask for help.

If your program works without complaint, that means that your import worked, and you now have access to Processing functionality!

To get Processing to work, we need to do a whole bunch of things. As this is unfortunately complex and requires things you haven't learned yet, my advice is to copy-paste my boiler-plate code below. However, at least try to read through the following changes:

- integrate your code into the Processing object model. Unfortunately this uses new object oriented programming techniques that you haven't learned. You need to make your class *extend* the existing Process class;
- modify `main` to get it to start the Processing engine, which unfortunately uses an advanced technique called *reflection* that you haven't learned yet;
- put your regular Processing functions inside this new class, and also put the word `public` in front of them, for object-oriented reasons you haven't learned yet;
- Put your `size` command (to set the canvas size) not in the `setup`, but instead in a new function called `settings`.

If you do all this, your program should work. Since you are extending the `PApplet`, your familiar Processing commands are freely available within the class.

Here is the boiler plate:

```
import processing.core.PApplet;

public class FirstProgram extends PApplet
{
      public void settings()
      {
            size(500,500);
      }

      public void setup()
      {
      }

      public void draw()
      {
      }

      static public void main(String[] args) {
         String[] appletArgs =
                new String[] {"FirstProgram"};
         PApplet.main(appletArgs);
        }
}
```

The only gotcha here is that the string used in the main MUST match the name of your class. That's a little annoying, but can't be helped.

Also, there is likely some confusion as to how to integrate other concepts you have learned, so here is a broader example including arrays, user defined functions, etc.

```
import processing.core.PApplet;

public class FirstProgram extends PApplet
{
	final int POINTS = 100;
	final int BEAM_SIZE = 50;
	final int SIZE = 2;
	float[] x = new float[POINTS];
	float[] y = new float[POINTS];
	public void settings()
	{
		size(500,500);
	}

	public void setup()
	{
		for (int i = 0; i < POINTS; i++)
		{
			x[i] = random(width);
			y[i] = random(BEAM_SIZE)-BEAM_SIZE/2;
		}
	}

	public void drawBeam()
	{
		for (int i = 1; i < POINTS; i++)
		{

		  ellipse(x[i]+mouseX, y[i]+mouseY,SIZE,SIZE);
		  x[i] = (x[i]*x[i]+y[i])%width;
		}
	}

	public void draw()
	{
		background(0);
		stroke(255);
```

```
            drawBeam();
      }

      static public void main(String[] args) {
         String[] appletArgs =
                  new String[] {"FirstProgram"};
         PApplet.main(appletArgs);
        }
}
```