**Keyboard and Mouse Interaction in Processing**

**Carson Foster**

Now we start to get to some of the more interesting material. In this lesson, we’ll be discussing how to interact with the user’s keyboard and mouse. This will allow you to make more complex, more intriguing, and more useful applications in Processing.

Mouse Interaction

We will start off by discussing the ways of monitoring the state of the mouse in the normal draw function. We haven’t really used this function, since our programs have been static images that don’t change. Recall that the draw function runs once every frame (and the default is 60 frames per second). This section is differentiated from events/listeners, which we will discuss in a later subsection. Let’s look at the different types of information we can extract from the mouse.

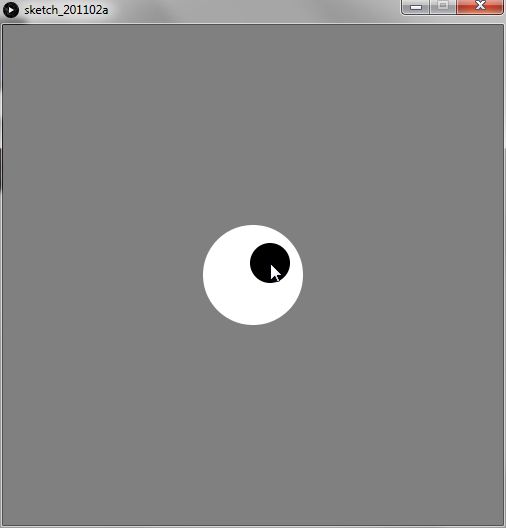
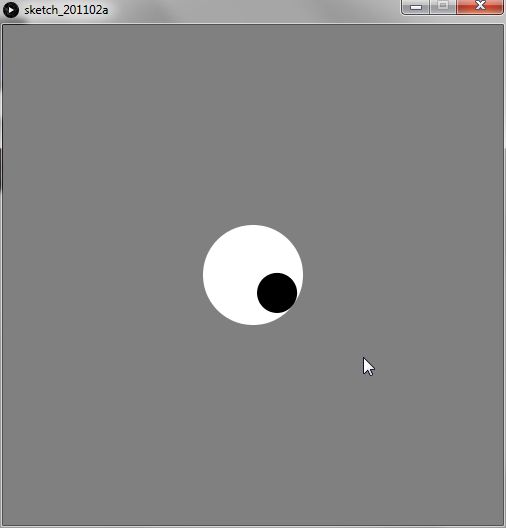
First, we can find the coordinates of the mouse cursor using two special variables:



These variables act like any other float variable, but change after each call to the draw function. The coordinate system they use is always the exact same as the default Processing coordinate scale, so they are not affected by transformations.

Next, we can also look at the coordinates of the mouse cursor on the previous frame, which allows us to make comparisons with the coordinates from this frame. These values are stored in the pmouseX and pmouseY variables.

Using just this information, we can make some interesting programs. For example, we could make an Eye class that simply draws an eye that looks wherever the cursor is.



Try this code out for yourself! The code is mostly

self-explanatory, but the atan2 function returns

the angle (in radians) for a given set of (y, x) coordinates: it acts as an inverse tangent function that is given the side lengths of the triangle instead of a value. Here are some example screenshots, but the code can be more fully appreciated when you run it yourself.

Next, we can also poll the mouse for the status of its buttons: whether a button is pressed, and if so, which button is pressed.