Reacting to the Mouse and Keyboard

Mike Bailey
mjb@cs.oregonstate.edu
The *mousePressed*, *mouseX*, and *mouseY* Variables

```java
void draw( )
{
    stroke( 0, 0, 0 );
    fill( 255, 50, 50 );
    if( mousePressed )
    {
        Circle( mouseX, mouseY, 50, 20 );
    }
}
```

*mousePressed* is a built-in variable that is always telling you if a mouse button is currently pressed.

*mouseX* and *mouseY* are built-in variables that are always telling you where the mouse cursor is.
The `mousePressed`, `mouseX`, and `mouseY` Variables

```java
void draw() {
    stroke( 0, 0, 0 );
    fill( 255, 50, 50 );
    if( mousePressed )
    {
        Circle( mouseX, mouseY, 50, 20 );
    }
}
```
The *mousePressed*, *mouseX*, and *mouseY* Variables
The `keyPressed` and `key` Variables

```java
void draw( )
{
    if( keyPressed )
    {
        switch( key )
        {
            case 'r':
                fill( 255, 50, 50 );
                break;
            case 'g':
                fill( 50, 255, 50 );
                break;
            case 'b':
                fill( 50, 50, 255 );
                break;
            
        }
    }
    
    if( mousePressed )
    {
        Circle( mouseX, mouseY, 50, 20 );
    }
}
```

- The `stroke()` and `fill()` calls have been moved to `setup()`
- `keyPressed` is a built-in variable that is always telling you if a keyboard key has been pressed
- `key` is a built-in variable that tells you what key has been hit
- The `switch/case` statements are Processing's way of checking many values without having a whole slew of if-statements
The `keyPressed` and `key` Variables

```cpp
void draw() {
    // moved the stroke() and fill() calls to setup()
    if (keyPressed)
    {
        switch (key)
        {
            case 'r':
                fill(255, 50, 50);
                break;

            case 'g':
                fill(50, 255, 50);
                break;

            case 'b':
                fill(50, 50, 255);
                break;
        }
    }
}

if (mousePressed)
{
    Circle(mouseX, mouseY, 50, 20);
}
```
You can also define your own functions to handle the mouse and keyboard explicitly, but we don’t need these yet.

```java
void mousePressed( )
{
    if( Debug )
        println( "mouse button = " + mouseButton );
}

void mouseMoved( )
{
    if( Debug )
        println( "mouse has been moved: " + mouseX + ", " + mouseY );
}

void mouseDragged( )
{
    if( Debug )
        println( "mouse has been dragged: " + mouseX + ", " + mouseY );
}
```