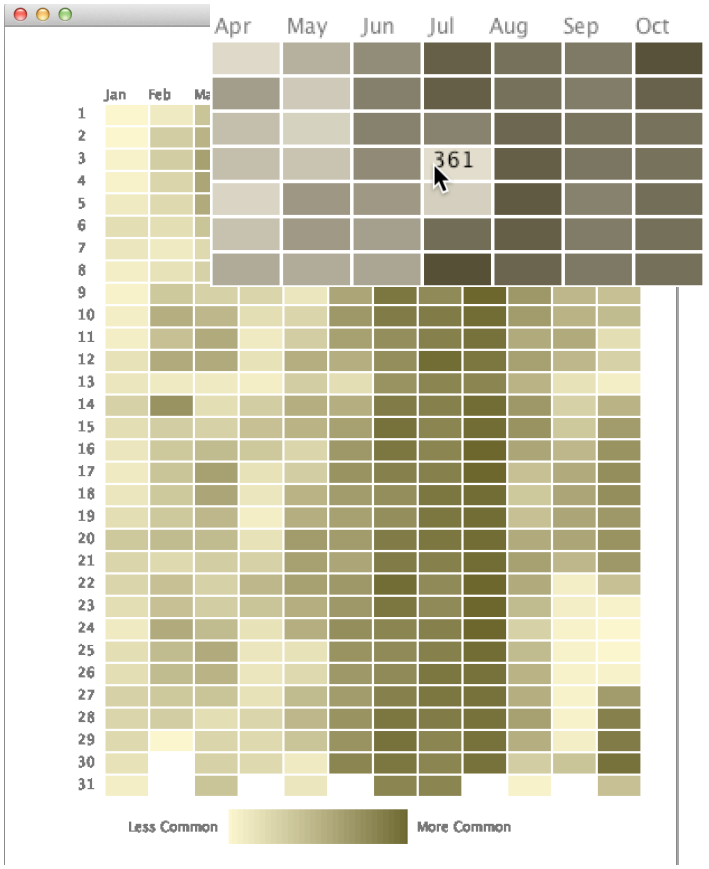
Lab Exercise 10: Birthday Hover

Goal

The goal is to practice with arrays and track user behavior to provide more specific information.

Starting Point

At this point you have completed assignment 14, the Birthday Infographic. The plan now is to allow users to hover the mouse over a rectangle, and to print out the ranking data where the mouse is. This will require just one function of about 10 lines, so there is very little programming. There *is* some thinking.



Review Some Numbers You Used

In writing your Birthday Infographic you used the following numbers:
a) Rectangles are (38, 18) but we round to (40, 20).
b) January is column 1, December is 12.
c) The starting point (upper left corner) of the rectangles is (50, 50) *in your Warmup*, but because months and days are numbered starting at 1, the upper left corner of rectangles in the Infographic is (50+1*40, 50+1*20); check your code!

So, the mouse will be over January 1st if
50+1*40 < mouseX < 50+2*40 and
50+1*20 < mouseY < 50+2*20
and it's over February 1st if
50+2*40 < mouseX < 50+3*40 and
50+1*20 < mouseY < 50+2*20.
And so forth. From these numbers we figure out that the (month, day) index position is given by

```
int mIndex = (mouseX-50)/40;     // Figure number of month
int dIndex = (mouseY-50)/20;     // Figure number of day
```

Why is this right? Because in all cases, to figure which rectangle the mouse is over, we need to get rid of the "50"s. Once we've done that we need to skip by 40s going across and skip by 20s going down; so we divide by those numbers. The calculation will give a float number; for example suppose the mouse is near the center of January 1st at mouseX equal 113, and mouseY equal 81. Then mouseX-50/40 is 1.575 and mouseY-50/20 is 1.55. When these are made into integers – dropping the digits right of the decimal point – the result in mIndex is 1 and dIndex is 1, as we want.

The display( ) *Function*

Show the computation of the last section by writing a function display( ) that is called from draw( ) and declares two integers, mIndex and dIndex, as the first two lines inside of the function. (This is contrary to our usual rule of putting variable declarations at the top of the program.) These two are assigned the position calculation from above to figure the month and day indexes.

Show that this works by printing them out at the bottom of the screen with
println(mIndex + " " + dIndex);
As you move your mouse around the correct month and day should appear. Notice that it gives (erroneous) indexes for February 30th.

As you move the mouse to the edge of the canvas, you notice that some of the values are illegal days, like -1 and 33. Because we only want values that are in the table range, enclose the println( ) command in an if-statement verifying that 0 < mIndex < 13 and 0 < dIndex < 32. Now, the numbers should all be correct. Check it.

How Many Days Since Jan 1?

To print out the rank of a birthday, we need to know where it is in our list of 366 items in the rank[ ] array. Our problem is that each month has a different number of days, so we cannot just multiply it out like we did the pixel position in the Beyonce assignment. So, we solve the problem by creating a new array of 12 items, saying for month i how many days precede the first of the month.

Declare a new integer array dayTot after your other data arrays, and initialize it to these values,
{0, 31, 60, 91, 121, 152, 182, 213, 244, 275, 305, 335}
which are the total number of days preceding the 1st of the month. Thus, 0 days precede January 1st, 31 days precede February 1st, etc.

Referencing the rank[ ] *Array*

We are now ready to print out the rank value. For a given day, say July 4th, our mIndex equals 7 and dIndex equals 4, its rank at index 182+4-1. Where did that number come from? It's the dayTot value for July, plus the dIndex for Independence day minus 1, because we always index from 0. So, all we need is a simple
text(rank[…], mouseX, mouseY);
replacing the println( ) used in the last section; of course, there is stuff to fill in at the ellipsis.

Contrasting Text

All that remains is to set the fill( ) so that it is easier to read the rank value when it is displayed over the rectangle. For that, we simply use an if-statement.

So just above the text( ) function of the last section (that is, inside of the if-statement), place another if statement that tests the value about to be displayed, i.e. rank[…]. If the

birthday ranks in the first half of all birthdays in the year (`rank[…]` < 183), then fill using a light color, because the smaller rank values are dark; otherwise, display using a dark color.

The infographic is nearly perfect. It is wrong in that for shorter months like February that have days with no entry, it displays values that correspond to the first of the next month. This is easy to fix, but we will not do so today.

Wrap Up: You have initially assigned values to arrays, and then used those arrays to display birthday rank data as an infographic based on where the mouse is hovering.

Turn In: Place your completed, documented code in the lab 10 dropbox.