CSE120: Computer Science: Principles

Homework 8: Making Blinky Work

Goal: In this assignment you will extend a program that you developed earlier, making it "work" with mouse clicks and mouse motions. This will require that you to do four things:

- Make the program Active
- Make Blinky move right (you've done "move right" before) and left
- Add a `mousePressed( )` block
- Control Blinky using the position of the mouse

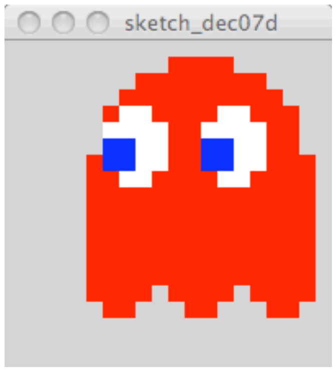
The steps are described below. (You may wish to review the Assignment Resource.)

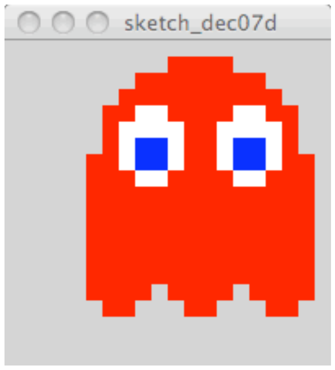
Recap

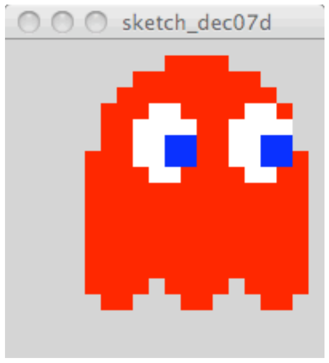
In the last assignment (07) you programmed the Blinky ghost from the ancient game of Pacman. In addition to drawing the image, you worked out how to move the eyes. Doing so required that you add a declaration for the integer `look`, and modify the code for the white and blue rectangles so that they were positioned using a computation of the form

```
rect( 60 + (look*10), 50, 10, 30);
```

By changing the value that you used to initialize `look`, you were able to get it to look left, center or right. That's where we are now.







Activating Blinky

We will make Blinky move. To prepare for that, we need to make our program active, which means (you'll recall) that we add the `void setup( ) { … }` and `void draw( ) { … }` blocks. Position these after the declaration for `look`. Put the `size( )` and `noStroke( )` commands in the `setup( )` block, and the other drawing commands in the `draw( )`. Finally, lets make the background black, like it's supposed to be for this game, by adding the `background( )` command at the start of the `draw( )` block.

Check it to be sure that you got what you had before, but with a black background.

Moving Blinky Right

Recall from Assignment 05 that you moved the robot right by adding `x` to the x-position parameter of every one of the rectangles. That should work here, too. The steps you used in Assignment 05 – after we remove intermediate steps of moving each part separately – were

- Before `setup( )` declare an integer `x` initialized to 0
- Add `x=x+1;` as the last statement of the `draw( )` block
- Add `+x` to the first (x-position parameter) of all of the `rect( )` function calls

Notice that you need to move the eyes left, too, even though you are already adding `+(look * 10)` or `+(look * 20)` to the x-position parameter to reposition them. Those should stay so that Blinky can still look left, center or right when it moves (though he will only do one of those at this point).

After you have added `x` to each of the x-position parameters for every rectangle, test it out to see it move across the screen. (Do you want to make the canvas width wider? Do you want `x=x+2` instead?)

Moving Blinky Left

Change your program so that Blinky moves left.

Once you have Blinky moving left, declare an integer `dir`, and revise your program so that Binky moves right if `dir` is 1 and it moves left if `dir` is -1. A good way to work on getting this right is to initialize `dir` to 1 in the declaration, and work out the logic of making Blinky move based on `dir`. While you're working on this logic, the ghost will only go right. Then, change the initialization to -1 and see it will go left.

Control Using the Mouse

Now we want to control which way Blinky's moving by where the mouse is positioned. If the mouse is left of Blinky's centerline, it goes left; if the mouse position is to the right of its centerline, it goes right.

This change will be made at the end of the `draw( )` block, because that is where the `x` value that moves Blinky is set. To make the change we will need an if-statement that tests where the mouse is in relation to where Blinky is and sets the `dir` variable. Thinking it through, if `mouseX` (recall `mouseX`?) is smaller than Blinky's centerline then the mouse is to its left and it should go left. But if `mouseX` is larger than Blinky's centerline, then it should go right.

What value does Blinky's centerline have? If Blinky is 14 bars wide then it must be the x-position used to draw the 8th bar, because the 8th bar is the first one on the right hand side of Blinky, if you drew them left-to-right as instructed in Assignment 07, and the bar's x-position is where that bar starts, that is, in the middle. Of course, because Blinky's moving, the value will also include the addition of `x`, as in, `<`*some number*`>+x`.

Finally, the if-statement's test will be followed by two parts – the *then*-part where we do commands if the outcome of the test is true, such as setting `dir` properly, and the *else*-part (preceded by else) where we do the commands if the outcome is false, such as setting `dir` the other way. So, give it a try!

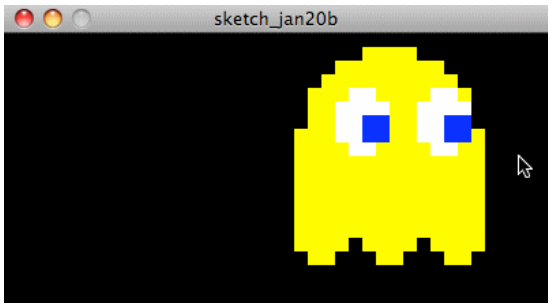
Watch Where You're Going

Blinky can move right and left, but it doesn't always look where he's going. Obviously, it needs to change the direction that its eyes are looking. You can fix this at the same time that you set `dir` to make Blinky go in the correct direction. Just set the `look` variable. You can set it so Blinky looks where its going or looks where it came from. Your choice.

Changing Color

The last improvement to the program is to change the color between red and some other color, say light yellow: `color(255,255,128)` because there was another ghost that was light yellow. Obviously, to do that you will need to declare a `color` variable at the top (with `x` and `dir`) and you will need to initialize it to the red settings. Then, the fill command that defines Blinky's color needs to use this variable. Finally, you need to add a `mousePressed( )` block, and in the block set the color variable to either red or yellow. You could set it as a one-time change, but it is more fun to be able to switch back and forth between red and yellow mouse clicking. For that, you will need one more `int` variable that remembers what color you will set Blinky to next time. (Will you encode red next time as 0 and yellow next time as 1?) Of course, the variable must be declared and initialized at the top of the program, and in `mousePressed( )` you will need to use it in an if-statement to set the color and change the `int` value for next time.

Save a copy of the `.pde` file, renamed <*your name*>`.pde`.



Wrap Up

Blinky is now animated and he can move left or right according to where the mouse is located. He looks in a standard direction. Also, when the mouse is clicked, he can change color.

Turn In

Be sure that the changes you made in this assignment are fully documented with comments explaining what you are doing. IN THIS ASSIGNMENT COMMENTS WILL BE GRADED. Turn in the file into the course drop box.