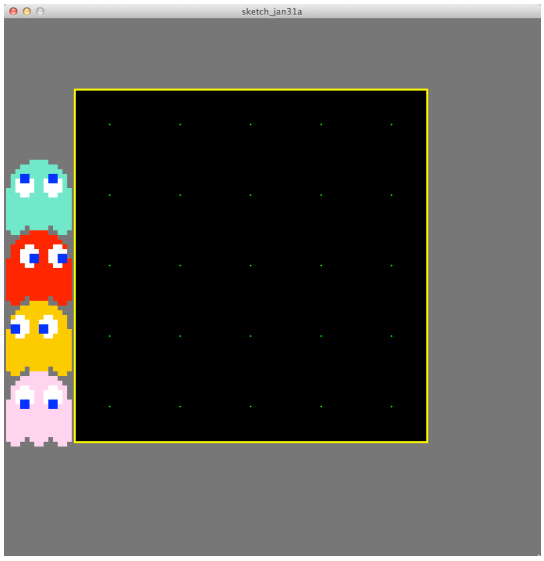
Homework 11: Controlling Blinky

Goal: The purpose of this exercise is first to use a program you have already written, and second to gain experience controlling interactive applications.

In this assignment you will build a simple application using the Blinky program from assignment 8. Using components built previously to create more complex programs is a fundamental way to build software. Follow these steps.

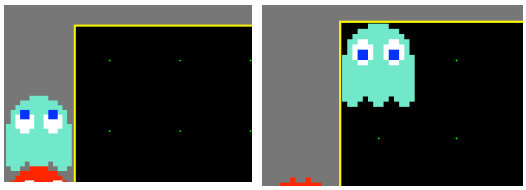
Step 1: The Display. Using your Blinky functions and the game board code supplied with the assignment, create the initial display. This should be drawn in setup( ) because it should not be redrawn continually. Looking at the display



you will create an 800x800 canvas with gray background, draw four copies of your Blinky ghost, with colors and eyes directed as shown, and draw the game board.

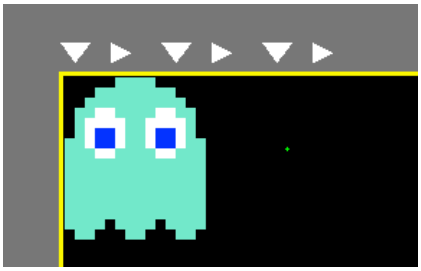
Notice from the game board code that the board is 750, 765 and it starts at 150,150. This means that its lower right corner is at 900x915, which is off the canvas. So you *must* include a scaling command in both your setup( ) and draw( ) code using scale(0.7).

Step 2: Set Blinky. The player of the game will be allowed to choose which ghost to play with. These are chosen by typing one of the keys, 1, 2, 3, 4. In this step make a keyPressed ( ) function that uses four if-statements to check for the keys, and remembers the proper ghost color. The chosen ghost should be erased. It should be redrawn in the corner of the game board with the eyes looking straight forward. To erase the ghost, redraw its body with background color. At right are before/after images resulting form the user typing the key '1'.



Hint: You will need a global color variable to remember the ghost color; if you initialize it to black, then the ghost can be drawn in draw( ) as soon as the ghost color has been chosen, that is, when it's not black any more, which can be checked with an if-statement.

Step 3: Add Commands. The ghost will move only right and down. To move the ghost the user will type one of two letters; I will use 'j' for down, and 'k' for right, but you can choose any letters. Add to your keyPressed( ) function to include if-statements for the two letters. When the user types a letter, a white triangle displays at the top of the game board pointing in the direction of motion. At right is the result of my typing: j k j k j k



Hint: To print the triangles so they have their own space as the user types, I declared an integer, nextTri and initialized it to 150. With each letter, I used that value to compute the x-positions for the triangle, and then added 40 to it to set up for the next triangle.

Step 4: Animate The Ghost. To make the ghost move, you will write an animate( ) function and call it from draw( ), replacing the original logic that draws the ghost when the color is changed. To do this, animate( ) will need four more global integers declared: xpos and ypos, both initialized to 155, and xmove and ymove, both initialized to 0.

Here's the idea: To move the ghost the user types a letter. That letter will be recognized by keyPressed( ), the triangle will be drawn, and *150 will be added to either xmove or ymove depending on which direction the ghost is supposed to move*. The 150 is the amount it is to move in the chosen direction. (So, we have to make two more mods to our keyPressed( ) function to increase xmove or ymove by 150.)

The animate( ) function will erase the ghost wherever it is by drawing its body in black at xpos, ypos. It will even erase it if the user hasn't actually chosen the color yet. (This erases the upper left dot, but that's OK for now.) Then, using an if-statement, animate( ) checks to see if xmove is greater than 0, and if it is, it adds 1 to xpos to move the ghost, and subtracts 1 from xmove to keep track of how much further it has to go; it uses another if-statement to do the same for ymove and ypos. Finally, with one more if-statement, animate( ) checks to see if the color has changed, and if it has, it draws the ghost at xpos, ypos in the proper color. So, the logic of animate( ) is

Erase the ghost as xpos, ypos
Check if it has farther to move in the x-direction, and if so add to xpos
Check if it has farther to move in the y-direction, and if so add to ypos
Draw the ghost at xpos, ypos (assuming the user has chosen the color)

Wrap up. You've used your previous code in a new application to control the screen.

Turn In. Turn in the program to the course dropbox.