1H – Animation using image frames

1. Start with one of the finished products that has double buffering in it. (If it has another animation on it like the cats or a smiley, all the better)
2. Perform an animation using the image <xBallsTrans.gif> as follows. (xBalls.gif is the non-transparent version of the same gif).
   1. There should be two inputs to your algorithm: (in other words variables that I can change so I can recompile and rerun and see the results):
      1. **Initial revolutions per second:** should be a double that represents the initial rotational velocity of the ball in revolutions per second. (2 would be 2 revs per second, 0.25 would be a quarter of a rev per second).
      2. **Deceleration (in revolutions per second squared):** should be a double that represents the number of revolutions per second lost per second. (0 would represent no slowing down, 0.1 would represent a per second decrease of one tenth of a revolution per second, 0.01 would represent a per second increase of one hundredth of a revolution per second). The equations to handle this should be similar to linear motion (except instead of working in pixels per second, we’ll be working in revolutions per second).
   2. I suggest you start by just getting the ball to spin at a constant velocity (revs per second) before you put deceleration logic in.
   3. Use an elapsed timed technique to calculate the elapsed time for each iteration through the loop and use that to calculate how many times the ball should spin in that time slice. It’s a modulus arithmetic operation on that result to get the image number to display.
   4. If the instantaneous velocity is less than zero, do not alter the ball that is displayed. If you declare the ball number to display as a class level instance variable, then the last ball number displayed will be remembered in this variable.