Q1)

The problem was relatively straightforward and I broke it into the following parts:

* **Animation/double buffering**

I started with first learning how to animate a single square without screen flickering.

To do this I referenced the book and followed their example, I having the x position depend on alpha.

* **Drawing the wheel and moving it**variables:
  + Hub Radius
  + Inner Radius
  + Outer Radius

Drawing wheel consisted of filling 3 circles in this order: outer circle with radius of Outer radius, inner circle with radius of the inner circle on top of the previous circle then the hub is drawn on top of all of that.

Drawing the spokes was a little more complicated, to get 6 evenly spaced spokes I first draw a line from the middle of the wheel to a point that is the Inner Radius distance horizontally from the middle. Then I calculate 5 more points one at a time rotating by 60 degrees about the center from each other and draw a line from the center to each of those points to get the rest of the spokes. I drew the spokes before the hub so that the hub covered the intersection of the spokes.

To move the wheel the wheel I adjusted all drawn point’s x coordinates by the alpha for the animation.

* **Rotating the wheel based on the position (avoid slipping)**

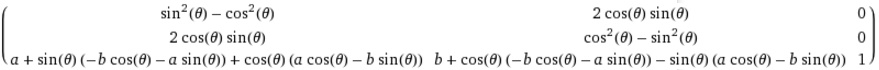
To rotate the wheel I calculate the angle to rotate the spokes by depending on how far it has moved horizontally (alpha). The angle to rotate the wheels can be given by the equation where the radius is the outside radius of the wheel. Once I know the angle I can rotate the first spoke by theta then calculate the rest of the spokes by rotating by 60 degrees 5 times.

Q2)

Reflection about one of the axes can be done by just flipping either the x or y coordinates of the vector. Rotating about an arbitrary axis can be done by first translating the vector to the origin, rotate it to match one of the axes, and reflect about that axis. Once the reflection is calculated we then do the inverse of rotating it and translating it. The resultant point is the point reflected about the specified axis.

The reflected point (x’, y’) about a line specified by (a,b) and (c,d) can be computed by the matrix vector product:

(x’, y’, 1) = (x, y, 1)

The final matrix for this transform is:  
 

Where is the angle from the x axis of (c’, d’) = (c, d)

This transform is applied to every point in the polygon to get the reflected polygon.