In order to have the ball rebound correctly after hitting a brick, we need to determine which side of the brick is being hit.

The boundary of the ball intersects with the boundary of the brick before the center of the ball intersects with the brick. This means that we can test the forward motion of the ball against each of the brick sides to determine which side is being struck. To do this we can use a vector (arrow) computed from the ball’s center position and its velocity vector (dx,dy). We’ll call this the \textit{ball collision vector}.

We also need to determine the length of the ball’s collision vector. If it is too long, it will also intersect with the far side of the brick. Since the ball has just touched the brick, a length equal to the ball’s diameter would be long enough to ensure that the vector crosses one side, but doesn’t extend through the brick to the far side.

Here’s the calculations:

- Normalize the velocity vector to a length of 1:
  \[
  \text{velocity} = \text{sqrt(dx*dx+dy*dy)}; \\
  \text{normalized\_dx} = \text{dx/velocity}; \\
  \text{normalized\_dy} = \text{dy/velocity};
  \]

- Scale the velocity vector to a length equal to the diameter of the ball (twice the radius):
  \[
  \text{test\_dx} = 2.0*\text{radius}\times\text{normalized\_dx}; \\
  \text{test\_dy} = 2.0*\text{radius}\times\text{normalized\_dy};
  \]

- The collision vector is now the line segment from \((x,y)\) to \((x+\text{test\_dx},y+\text{test\_dy})\)

As we mentioned in class, the intersection functions in \texttt{java.awt.geom.Line2D} can be used to test for line intersections. See \url{http://java.sun.com/j2se/1.4.2/docs/api/java/awt/geom/Line2D.html} for details.