Lab 6

Step 1:
Use the Game, Ball, and Cannon.java files developed during class. Modify the Game
class so that instead of using random variables you ask the user for the initial X and
Y velocities of the ball.

Step 2:
Create a TargetCircle class. The TargetCircle will appear on the screen as a filled
MyroCircle. It should be somewhere on the bottom of the screen. This TargetCircle
should have an intersects() method, which will take in an x, y, and radius as
parameters and return true if the circle described by x,y,radius intersects the
TargetCircle.

Step 3:
Create a direction() method in TargetCircle. When passed an x,y,radius triple, the
method will return the x and y vector from the center of the TargetCircle to the
center of the passed in circle.

Step 4:
Create an explode() method. This method will cause the TargetCircle to disappear. If
you’d like, you can add an animation to make it flashy.

Step 5:
Have the game create a new TargetCircle object and then ask the user for initial X
and Y velocities. Modify the Ball class so that its constructor also takes in a
TargetCircle as an input parameter. If the Ball intersects the TargetCircle, have the
Ball bounce away from the TargetCircle in the direction from step 3 and make the
TargetCircle explode.

Step 6:
Create a new class, TargetSquare, which has the same methods as TargetSquare.
Instead of bouncing away in the direction for step 3, the ball should bounce against
the three possible sides.

Step 7:
Create a Target interface. What methods does it need?

Step 8:
Make both TargetSquare and TargetCircle implement the Target interface. Now,
instead of passing the TargetCircle as a parameter to the Ball class, pass a Target.