Breakout Phase 3

In which we add walls to contain the ball and colored bricks to break. You may wish to consult the images for classic Breakout on Wikipedia.

1. Save your `breakout2.py` as `breakout3.py` and modify as follows.
2. Add a `Wall` class with two methods:
   - `__init__(self, win, p1, p2)` -- creates a rectangular wall in `win` with `Points` `p1` and `p2` as opposite corners. The critical instance variable for the wall is a `BoundingBox` to handle collisions.
   - `bounce(self, ball)` -- bounces the ball off the wall. Returns a Boolean indicating whether a bounce occurs.
3. Modify the `BreakoutGame` application to test your `Wall` class.
   - Remove the old `obstacle` instance variable and the `if` statements that cause the ball to reflect at the window edges.
   - Make the Window 1024x768 and add an instance variable, `walls`, that holds a Python list of 4 `Wall` objects near the edges of the window to contain the ball. Later on, we'll replace the bottom wall with a paddle. Make sure that you adjust the window coordinates to match the aspect ratio of the new window (4:3) so that the ball stays round.
   - Modify the `run()` method so that the ball bounces off the walls. For example, you might add something like:

     ```
     for wall in self.walls:
         if wall.bounce(ball):
             print("BOUNCE!")
     ```

   - Test/debug your bounded bouncing ball before continuing; you may need to adjust things like the `update` timing, ball radius, or move size to accommodate your new window size.
4. Add a `Brick` class with methods similar to the `Wall` class. `Brick` is like `Wall`, except for two minor modifications:
   - The constructor for `Brick` takes the color of the brick as another parameter.
   - The `bounce` method should undraw the `Brick` if it is hit.

5. Update the application to include a list of `Bricks` with random locations, sizes, and colors. Update the animation loop so that it checks for brick bounces and removes hit bricks from the list. Have the animation continue until all the bricks are gone. Place the initial ball and bricks to ensure the ball does not start out trapped inside of a brick. Hint: you can generate a random color with a line of code like this:

```
color = color_rgb(randrange(256), randrange(256), randrange(256))
```