

Bouncing Ball Example

Based on a handout by Patrick Young.

```
/*
 * File: BouncingBall.java
 * -----
 * This program graphically simulates a bouncing ball.
 */

import acm.program.*;
import acm.graphics.*;

public class BouncingBall extends GraphicsProgram {

    // size of ball
    private static final int BALL_DIAMETER = 30;

    // initial X and Y location of ball
    private static final double X_START = BALL_DIAMETER / 2;
    private static final double Y_START = 100;

    // pause time between ball moves
    private static final int ANIMATION_DELAY = 50;

    // horizontal velocity (how fast ball moves to right)
    private static final double X_VELOCITY = 5;

    // amount Y velocity is increased each cycle by gravity
    private static final double GRAVITY = 3;

    // fraction of Y Velocity reduced by a bounce
    private static final double BOUNCE_REDUCE = 0.1;

    public void run() {
        createBall();

        // End simulation when ball goes off to right of screen
        while (ball.getX() < getWidth()) {
            moveBall();
            if (ballHitFloor()) {
                bounceBall();
            }
            pause(ANIMATION_DELAY);
        }
    }

    /**
     * Create and place ball.
     */
    private void createBall() {
        ball = new GOval(X_START, Y_START, BALL_DIAMETER, BALL_DIAMETER);
        ball.setFilled(true);
        add(ball);
    }
}
```

```

}

/**
 * Update Y velocity due to gravity, and move the ball.
 */
private void moveBall() {
    yVelocity += GRAVITY;
    ball.move(X_VELOCITY, yVelocity);
}

/**
 * The ball has hit the floor if it is no longer above the
 * floor.
 */
private boolean ballHitFloor() {
    return distanceAboveFloor() <= 0;
}

/**
 * Calculate the distance between the bottom of the
 * ball and the floor (i.e., the bottom of the window).
 *
 * A negative distance means that the ball has dropped
 * below the floor.
 */
private double distanceAboveFloor() {
    double ballBottom = ball.getY() + BALL_DIAMETER;
    return getHeight() - ballBottom;
}

/**
 * Update the ball's Y velocity to reflect moving in
 * the opposite direction, and move the ball back above
 * the floor (in case it dropped below).
 */
private void bounceBall() {
    // change ball's Y velocity to now bounce upwards
    // NOTE: won't take effect until next moveBall()
    yVelocity = -yVelocity * (1 - BOUNCE_REDUCE);

    // distanceAboveFloor is negative when ball is below
    double distanceBelowFloor = -1 * distanceAboveFloor();

    // move ball above the floor by the same amount it
    // dropped below (-1 instead would rest ball on floor)
    ball.move(0, -2 * distanceBelowFloor);
}

/* Private instance variables */
private GOval ball;
private double yVelocity = 0.0;
}

```