```
import acm.graphics.*;
import acm.program.*;
import acm.util.*;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class Breakout extends GraphicsProgram
{

    /** Width and height of application window in pixels */
    public static final int APPLICATION_WIDTH = 400;
    public static final int APPLICATION_HEIGHT = 600;

    /** Dimensions of game board in pixels (usually the same) */
    private static final int WIDTH = APPLICATION_WIDTH;
    private static final int HEIGHT = APPLICATION_HEIGHT;

    /** Dimensions of the paddle */
    private static final int PADDLE_WIDTH = 60;
    private static final int PADDLE_HEIGHT = 10;

    /** Offset of the paddle up from the bottom */
    private static final int PADDLE_Y_OFFSET = 30;

    /** Number of bricks per row */
    private static final int NBRICKS_PER_ROW = 10;

    /** Number of rows of bricks */
    private static final int NBRICK_ROWS = 10;

    /** Separation between bricks */
    private static final int BRICK_SEP = 4;

    /** Width of a brick */
    private static final int BRICK_WIDTH =
        (WIDTH - (NBRICKS_PER_ROW - 1) * BRICK_SEP) / NBRICKS_PER_ROW;

    /** Height of a brick */
    private static final int BRICK_HEIGHT = 8;

    /** Radius of the ball in pixels */
    private static final int BALL_RADIUS = 10;

    /** Offset of the top brick row from the top */
    private static final int BRICK_Y_OFFSET = 70;

    /** Number of "lives" (balls) before the player loses */
    private static final int NUM_LIVES = 3;
```

```
    /** Global variables declared here.  You should feel free to add others as
needed. */
    int numRows = readInt("How many rows do you want. Must be between 10 and 20");
    int speed = readInt("Do you wan your speed to be level 1, 2, 3");
    GRect paddle;
    GOval ball;
    double vx;
    double vy;
    Color myColor = Color.red;
    int numBricks = (NBRICKS_PER_ROW) * (numRows);
    GLabel label;
    GLabel Wlabel;
    int numLives = NUM_LIVES;
    double vyMultiplier = 0;
    AudioClip bounceClip = MediaTools.loadAudioClip("bounce.au");

    /** Runs the Breakout program. */
    public void run()
    {
        settings();
        setupPaddle();
        setupBricks();
        setupBall();
        waitForClick();
        animationLoop();

    }

    public void settings(){

        if (numRows <10){
            numRows = 10;
        }
        if (numRows > 19){
            numRows = 19;
        }


    }

    public void setupBricks(){
        for(int i = 0; i < numRows; i++){
            for(int j = 0; j < NBRICKS_PER_ROW ; j++){
                GRect rect = new GRect(0 + (j *(BRICK_WIDTH+ BRICK_SEP))  ,
BRICK_Y_OFFSET + (i * (BRICK_HEIGHT+ BRICK_SEP)) , BRICK_WIDTH , BRICK_HEIGHT);
                rect.setFilled(true);

                if ((i  == 0) || (i == 1))
                {
                    myColor = Color.red:
```

```
                }
                if((i == 2) || (i == 3))
                {
                    myColor = Color.orange;
                }
                if((i == 4) || (i == 5))
                {
                    myColor = Color.yellow;
                }
                if((i == 6) || (i == 7))
                {
                    myColor = Color.green;
                }
                if((i == 8) || (i == 9))
                {
                    myColor = Color.cyan;
                }
                if(i > 9){
                    myColor = Color.blue;
                }
                rect.setColor(myColor);
                add(rect);

            }
        }

    }

    public void setupPaddle(){
        paddle = new GRect(0 + APPLICATION_WIDTH/2 - PADDLE_WIDTH/2,
APPLICATION_HEIGHT - PADDLE_Y_OFFSET - PADDLE_HEIGHT, PADDLE_WIDTH, PADDLE_HEIGHT);
        paddle.setFilled(true);
        add(paddle);


    }
    public void mouseMoved(MouseEvent event)
    {
        paddle.setLocation(event.getX() - PADDLE_WIDTH/2, APPLICATION_HEIGHT -
PADDLE_Y_OFFSET - PADDLE_HEIGHT );
        if (paddle.getX() >= APPLICATION_WIDTH - PADDLE_WIDTH){
            paddle.setLocation(APPLICATION_WIDTH - PADDLE_WIDTH, APPLICATION_HEIGHT -
PADDLE_Y_OFFSET - PADDLE_HEIGHT);
        }
        if (paddle.getX() <=  0){
            paddle.setLocation(0, APPLICATION_HEIGHT - PADDLE_Y_OFFSET -
PADDLE_HEIGHT);
        }
    }

    public void setupBall(){
```

```
        ball = new GOval(APPLICATION_WIDTH/2 - BALL_RADIUS, APPLICATION_HEIGHT/2 -
BALL_RADIUS/2, BALL_RADIUS*2, BALL_RADIUS*2);
        ball.setFilled(true);
        add(ball);
        vx = 1+Math.random()*2;
        if (Math.random()>0.5){
            vx = -vx;
        }
        vy = 3.0;

    }

    public void animationLoop(){

        while (true)
        {
            pause(5);
            updateBall();
            checkForCollisions();
        }

    }

    public void updateBall(){
        pause(5);
        if (speed == 1){
            vyMultiplier = 1.0;
        }
        if (speed == 2){
            vyMultiplier = 2.0;
        }
        if (speed == 3){
            vyMultiplier = 2.3;
        }
        if (speed > 3 || speed < 1){
            vyMultiplier = 2.3;
        }
        ball.move(vx, vy * vyMultiplier);
        if (ball.getX() > getWidth() - ball.getWidth() )
        {
            vx = -vx;
        }
        if (ball.getY() > getHeight() - ball.getHeight() )
        {
            if (numLives > 1) {
                ball.setLocation(APPLICATION_WIDTH/2 - BALL_RADIUS,
APPLICATION_HEIGHT/2 - BALL_RADIUS/2);
                pause(1000);
                numLives -= 1;
            }
```

```
            else{
                endgame();

            }
        }
        if (ball.getY() < 0)
        {
            vy = -vy;
        }
        if (ball.getX() < 0)
        {
            vx = -vx;
        }
    }

    public void checkForCollisions(){
        GObject object =  getElementAt(ball.getX() , ball.getY());
        if (numBricks == 0){
            remove(ball);
            remove(paddle);
            Wlabel = new GLabel("You Win", 0, 0);
            Wlabel.setFont("Times-Bold-48");
            Wlabel.setLocation(APPLICATION_WIDTH/2 - Wlabel.getWidth()/2,
APPLICATION_HEIGHT/2);
            add(Wlabel);
        }
        if(object == null){
            object =  getElementAt(ball.getX() + 2*BALL_RADIUS , ball.getY());
        }
        if(object == null){
            object =  getElementAt(ball.getX() + 2*BALL_RADIUS , ball.getY() +
2*BALL_RADIUS);
        }
        if(object == null){
            object =  getElementAt(ball.getX() , ball.getY() + 2*BALL_RADIUS);
        }
        if (object == paddle){
            if (numLives > 0){
                if (vy > 0)
                {
                bounceClip.play();
                vy = -vy;
                }

            }
        }



        else if (object == null){
```

```
        }

        else{
            if (numBricks > 0){
                remove(object);
                vy = -vy;
                numBricks -= 1;
            }
            else{

            }


            }

        }
    public void endgame(){
        remove(ball);
        remove(paddle);
        label = new GLabel("You Lose", 0, 0);
        label.setFont("Times-Bold-48");
        label.setLocation(APPLICATION_WIDTH/2 - label.getWidth()/2,
APPLICATION_HEIGHT/2);
        add(label);


    }


}
```