

```
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import javax.swing.Timer;
```

```
import javax.swing.JPanel;
```

```
public class GamePlay extends JPanel implements KeyListener, ActionListener {
    private boolean play = false;
    private int score = 0;

    private int totalBricks = 21;

    private Timer timer;
    private int delay = 8;

    private int playerX = 310;

    private int ballposX = 120;
    private int ballposY = 350;
    private int ballXdir = -1;
    private int ballYdir = -2;

    private MapGenerator map;
```

```
public GamePlay() {  
    map = new MapGenerator(3, 7);  
    addKeyListener(this);  
    setFocusable(true);  
    setFocusTraversalKeysEnabled(false);  
    timer = new Timer(delay, this);  
    timer.start();  
  
}  
  
public void paint(Graphics g) {  
  
    g.setColor(Color.white);  
    g.fillRect(1, 1, 692, 592);  
  
    map.draw((Graphics2D)g);  
  
    g.setColor(Color.yellow);  
    g.fillRect(0, 0, 3, 592);  
    g.fillRect(0, 0, 692, 3);  
    g.fillRect(691, 0, 3, 592);  
  
    g.setColor(Color.blue);  
    g.fillRect(playerX, 550, 100, 8);  
  
    g.setColor(Color.green);  
    g.fillOval(ballposX, ballposY, 20, 20);  
  
    g.setColor(Color.black);  
    g.setFont(new Font("serif", Font.BOLD, 25));
```

```
g.drawString("" + score, 590, 30);
```

```
if (totalBricks <= 0) {  
    play = false;  
    ballXdir = 0;  
    ballYdir = 0;  
    g.setColor(Color.green);  
    g.setFont(new Font("serif", Font.BOLD, 30));  
    g.drawString("You Won, Score: " + score, 190, 300);  
  
    g.setFont(new Font("serif", Font.BOLD, 20));  
    g.drawString("Press Enter to Restart.", 230, 350);  
  
}
```

```
if(ballposY > 570) {  
    play = false;  
    ballXdir = 0;  
    ballYdir = 0;  
    g.setColor(Color.red);  
    g.setFont(new Font("serif", Font.BOLD, 30));  
    g.drawString("Game Over, Score: " + score, 190, 300);  
  
    g.setFont(new Font("serif", Font.BOLD, 20));  
    g.drawString("Press Enter to Restart.", 230, 350);  
  
}
```

```
g.dispose();
```

```
}
```

```
@Override
```

```
public void actionPerformed(ActionEvent arg0) {  
    // TODO Auto-generated method stub  
    timer.start();  
    if(play) {  
        // Ball - Pedal interaction  
        if(new Rectangle(ballposX, ballposY, 20, 20).intersects(new  
Rectangle(playerX, 550, 100, 8))) {  
            ballYdir = - ballYdir;  
        }  
        for( int i = 0; i<map.map.length; i++) {  
            for(int j = 0; j<map.map[0].length; j++) {  
                if(map.map[i][j] > 0) {  
                    int brickX = j*map.brickWidth + 80;  
                    int brickY = i*map.brickHeight + 50;  
                    int brickWidth= map.brickWidth;  
                    int brickHeight = map.brickHeight;  
  
                    Rectangle rect = new Rectangle(brickX, brickY,  
brickWidth, brickHeight);  
  
                    Rectangle ballRect = new Rectangle(ballposX,  
ballposY, 20,20);  
  
                    Rectangle brickRect = rect;  
  
                    if(ballRect.intersects(brickRect) ) {  
                        map.setBrickValue(0, i, j);  
                        totalBricks--;  
                        score+=5;  
                    }  
                }  
            }  
        }  
    }  
}
```

```

+1 >= brickRect.x + brickRect.width)

        if(ballposX + 19 <= brickRect.x || ballposX

            ballXdir = -ballXdir;
        else {
            ballYdir = -ballYdir;
        }
    }

    }

    }

    }

    ballposX += ballXdir;
    ballposY += ballYdir;
    if(ballposX < 0) {
        ballXdir = -ballXdir;
    }
    if(ballposY < 0) {
        ballYdir = -ballYdir;
    }
    if(ballposX > 670) {
        ballXdir = -ballXdir;
    }

    }

    repaint();

```

```
}
```

```
@Override
```

```
public void keyTyped(KeyEvent arg0) {
```

```
    // TODO Auto-generated method stub
```

```
}
```

```
@Override
```

```
public void keyPressed(KeyEvent arg0) {
```

```
    // TODO Auto-generated method stub
```

```
    if(arg0.getKeyCode() == KeyEvent.VK_RIGHT) {
```

```
        if(playerX >= 600) {
```

```
            playerX = 600;
```

```
        } else {
```

```
            moveRight();
```

```
        }
```

```
    }
```

```
    if(arg0.getKeyCode() == KeyEvent.VK_LEFT) {
```

```
        if(playerX < 10) {
```

```
            playerX = 10;
```

```
        } else {
```

```
            moveLeft();
```

```
        }
```

```
    }
```

```
    if(arg0.getKeyCode() == KeyEvent.VK_ENTER) {
```

```

        if(!play) {
            play = true;
            ballposX = 120;
            ballposY = 350;
            ballXdir = -1;
            ballYdir = -2;
            score = 0;
            totalBricks = 21;
            map = new MapGenerator(3,7);

            repaint();
        }
    }

}

    public void moveRight() {
        play = true;
        playerX += 20;
    }

    public void moveLeft() {
        play = true;
        playerX -= 20;
    }

}

@Override
public void keyReleased(KeyEvent arg0) {
    // TODO Auto-generated method stub
}

}

```

```
import javax.swing.JFrame;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        JFrame obj = new JFrame();
```

```
        GamePlay gamePlay = new GamePlay();
```

```
        obj.setBounds(10, 10, 700, 600);
```

```
        obj.setTitle("Brick Breaker");
```

```
        obj.setResizable(false);
```

```
        obj.setVisible(true);
```

```
        obj.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        obj.add(gamePlay);
```

```
    }
```

```
}
```



```
import java.awt.BasicStroke;
```

```
import java.awt.Color;
```

```
import java.awt.Graphics2D;
```

```
public class MapGenerator {
```

```
    public int map [][];
```

```
    public int brickWidth;
```

```
    public int brickHeight;
```

```
    public MapGenerator(int row, int col) {
```

```
        map = new int [row][col];
```

```
        for (int i = 0; i < map.length; i++) {
```

```
            for (int j=0; j< map[0].length;j++) {
```

```
                map[i][j] = 1;
```

```
            }
```

```
        }
```

```
        brickWidth = 540/col;
```

```
        brickHeight = 150/row;
```

```
    }
```

```
    public void draw(Graphics2D g) {
```

```
        for (int i = 0; i < map.length; i++) {
```

```
            for (int j=0; j< map[0].length;j++) {
```

```
                if(map[i][j] > 0) {
```

```
                    g.setColor(Color.black);
```

```
                    g.fillRect(j*brickWidth + 80, i*brickHeight + 50, brickWidth,  
brickHeight);
```

```
                    g.setStroke(new BasicStroke(3));
```

```

        g.setColor(Color.white);
        g.drawRect(j*brickWidth + 80, i*brickHeight + 50,
brickWidth, brickHeight);
    }
}

}

}

public void setBrickValue(int value, int row, int col) {
    map[row][col] = value;
}

}

```

OUTPUT:

