package game;  
  
  
import java.awt.BasicStroke;  
 import java.awt.Color;  
 import java.awt.Graphics2D;  
  
public class mapGenerator  
{  
 public int map[][];  
 public int brickW;  
 public int brickH;  
  
 public mapGenerator (int r, int c)  
 {  
 map = new int[r][c];  
 for(int m = 0; m<map.length; m++)  
 {  
 for(int n =0; n<map[0].length; n++)  
 {  
 map[m][n] = 1;  
 }  
 }  
  
 brickW = 540/c;  
 brickH = 150/r;  
 }  
  
 public void drawgraph(Graphics2D gr)  
 {  
 for(int m = 0; m<map.length; m++)  
 {  
 for(int n =0; n<map[0].length; n++)  
 {  
 if(map[m][n] > 0)  
 {  
 gr.setColor(Color.*white*);  
 gr.fillRect(n \* brickW + 80, m \* brickH + 50, brickW, brickH);  
  
 *// this is just to show separate brick, game can still run without it* gr.setStroke(new BasicStroke(3));  
 gr.setColor(Color.*black*);  
 gr.drawRect(n \* brickW + 80, m \* brickH + 50, brickW, brickH);  
 }  
 }  
 }  
 }  
  
 public void setBrickVal(int v, int r, int c)  
 {  
 map[r][c] = v;  
 }  
}

package game;  
import java.util.\*;  
  
import java.awt.event.\*;  
  
import javax.swing.\*;  
  
import java.awt.\*;  
  
import javax.swing.Timer;  
public class Gameplay extends JPanel implements KeyListener, ActionListener  
{  
 private boolean Play = false;  
 private int scores = 0;  
  
 private int totalBrick = 48;  
  
 private Timer time;  
 private int delay=8;  
  
 private int playX = 310;  
  
 private int bposX = 120;  
 private int bposY = 350;  
 private int bXdir = -1;  
 private int bYdir = -2;  
  
 private mapGenerator map;  
  
 public Gameplay()  
 {  
 map = new mapGenerator(4, 12);  
 addKeyListener(this);  
 setFocusable(true);  
 setFocusTraversalKeysEnabled(false);  
 time=new Timer(delay,this);  
 time.start();  
 }  
  
 public void paint(Graphics gr)  
 {  
 *// background* gr.setColor(Color.*black*);  
 gr.fillRect(1, 1, 692, 592);  
  
 *// drawing map* map.drawgraph((Graphics2D) gr);  
  
 *// borders* gr.setColor(Color.*yellow*);  
 gr.fillRect(0, 0, 3, 592);  
 gr.fillRect(0, 0, 692, 3);  
 gr.fillRect(691, 0, 3, 592);  
  
 *// the scores* gr.setColor(Color.*white*);  
 gr.setFont(new Font("serif",Font.*ITALIC*, 25));  
 gr.drawString(""+scores, 590,30);  
  
 *// the paddle* gr.setColor(Color.*green*);  
 gr.fillRect(playX, 550, 150, 8);  
  
 *// the ball* gr.setColor(Color.*orange*);  
 gr.fillOval(bposX, bposY, 15,15 );  
  
 *// when you won the game* if(totalBrick <= 0)  
 {  
 Play = false;  
 bXdir = 0;  
 bYdir = 0;  
 gr.setColor(Color.*RED*);  
 gr.setFont(new Font("serif",Font.*ITALIC*, 30));  
 gr.drawString("You Won", 260,300);  
  
 gr.setColor(Color.*RED*);  
 gr.setFont(new Font("serif",Font.*ITALIC*, 20));  
 gr.drawString("Press (Enter) to Restart", 230,350);  
 }  
  
 *// when you lose the game* if(bposY > 570)  
 {  
 Play = false;  
 bXdir = 0;  
 bYdir = 0;  
 gr.setColor(Color.*RED*);  
 gr.setFont(new Font("serif",Font.*ITALIC*, 30));  
 gr.drawString("Game Over, Scores: "+scores, 190,300);  
  
 gr.setColor(Color.*RED*);  
 gr.setFont(new Font("serif",Font.*ITALIC*, 20));  
 gr.drawString("Press (Enter) to Restart", 230,350);  
 }  
  
 gr.dispose();  
 }  
  
 public void keyPressed(KeyEvent e)  
 {  
 if (e.getKeyCode() == KeyEvent.*VK\_RIGHT*)  
 {  
 if(playX >= 600)  
 {  
 playX = 600;  
 }  
 else  
 {  
 moveRight();  
 }  
 }  
  
 if (e.getKeyCode() == KeyEvent.*VK\_LEFT*)  
 {  
 if(playX < 10)  
 {  
 playX = 10;  
 }  
 else  
 {  
 moveLeft();  
 }  
 }  
 if (e.getKeyCode() == KeyEvent.*VK\_ENTER*)  
 {  
 if(!Play)  
 {  
 Play = true;  
 bposX = 120;  
 bposY = 350;  
 bXdir = -1;  
 bYdir = -2;  
 playX = 310;  
 scores = 0;  
 totalBrick = 21;  
 map = new mapGenerator(3, 7);  
  
 repaint();  
 }  
 }  
 }  
  
 public void keyReleased(KeyEvent e) {}  
 public void keyTyped(KeyEvent e) {}  
  
 public void moveRight()  
 {  
 Play = true;  
 playX+=20;  
 }  
  
 public void moveLeft()  
 {  
 Play = true;  
 playX-=20;  
 }  
  
 public void actionPerformed(ActionEvent e)  
 {  
 time.start();  
 if(Play)  
 {  
 if(new Rectangle(bposX, bposY, 20, 20).intersects(new Rectangle(playX, 550, 30, 8)))  
 {  
 bYdir = -bYdir;  
 bXdir = -2;  
 }  
 else if(new Rectangle(bposX, bposY, 20, 20).intersects(new Rectangle(playX + 70, 550, 30, 8)))  
 {  
 bYdir = -bYdir;  
 bXdir = bXdir + 1;  
 }  
 else if(new Rectangle(bposX, bposY, 20, 20).intersects(new Rectangle(playX + 30, 550, 40, 8)))  
 {  
 bYdir = -bYdir;  
 }  
  
 *// check map collision with the ball* A: for(int m = 0; m<map.map.length; m++)  
 {  
 for(int n =0; n<map.map[0].length; n++)  
 {  
 if(map.map[m][n] > 0)  
 {  
 *//scores++;* int brickX = n \* map.brickW + 80;  
 int brickY = m \* map.brickH + 50;  
 int brickW = map.brickW;  
 int brickH = map.brickH;  
  
 Rectangle rect = new Rectangle(brickX, brickY, brickW, brickH);  
 Rectangle ballrect = new Rectangle(bposX, bposY, 20, 20);  
 Rectangle brickrect = rect;  
  
 if(ballrect.intersects(brickrect))  
 {  
 map.setBrickVal(0, m, n);  
 scores+=5;  
 totalBrick--;  
  
 *// when ball hit right or left of brick* if(bposX + 19 <= brickrect.x || bposX + 1 >= brickrect.x + brickrect.width)  
 {  
 bXdir = -bXdir;  
 }  
 *// when ball hits top or bottom of brick* else  
 {  
 bYdir = -bYdir;  
 }  
  
 break A;  
 }  
 }  
 }  
 }  
  
 bposX += bXdir;  
 bposY += bYdir;  
  
 if(bposX < 0)  
 {  
 bXdir = -bXdir;  
 }  
 if(bposY < 0)  
 {  
 bYdir = -bYdir;  
 }  
 if(bposX > 670)  
 {  
 bXdir = -bXdir;  
 }  
  
 repaint();  
 }  
 }  
}

package game;  
  
import java.awt.Color;  
  
import javax.swing.JFrame;  
  
public class Main {  
 public static void main(String[] args) {  
 JFrame o=new JFrame();  
 Gameplay gamePlay = new Gameplay();  
  
 o.setBounds(20, 20, 700, 600);  
 o.setTitle("Breakout the Ball");  
 o.setResizable(false);  
 o.setVisible(true);  
 o.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 o.add(gamePlay);  
 o.setVisible(true);  
  
 }  
  
}