

```

package j4game;

import j4game.gui.Size;

import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.event.KeyEvent;

public class Tutorial implements Game{

    final static public int BALLS=100;
    static public GameController gameController;

    // x,y,x direction,y direction,size
    private double[][] balls=new double[BALLS][5];
    private Color[] ballsColor=new Color[BALLS];

    private Size gameSize;

    private boolean initDone=false;

    public Tutorial(){
    }

    private void init(){
        gameSize=gameController.getScreenSize();
        // initialise balls
        for (int t=0;t<balls.length;t++){
            // Set Location
            balls[t][0]=Math.random()*gameSize.getWidth();
            balls[t][1]=Math.random()*gameSize.getHeight();
            // Set Direction
            balls[t][2]=((Math.random()*2)+.5)*(Math.random()<.5f?-1:1);
            balls[t][3]=((Math.random()*2)+.5)*(Math.random()<.5f?-1:1);
            // Set Colour and Size
            balls[t][4]=Math.random()*5+2;
            ballsColor[t]=new Color(
                ((int)(Math.random()*127)+128),
                ((int)(Math.random()*127)+128),
                ((int)(Math.random()*127)+128));
        }
    }

    public void logic() {
        // Do initialisation
        if (!initDone){
            initDone=true;
            init();
        }

        // logic balls
        for (int t=0;t<balls.length;t++){
            // Move ball
            balls[t][0]+=balls[t][2];
            balls[t][1]+=balls[t][3];

            // Bounce ball
            if (balls[t][0]<0){
                balls[t][0]=0;
                balls[t][2]=-balls[t][2];
            } else if (balls[t][0]>gameSize.getWidth()-balls[t][4]){

```

```

        balls[t][0]=gameSize.getWidth()-balls[t][4];
        balls[t][2]=-balls[t][2];
    }
    if (balls[t][1]<0){
        balls[t][1]=0;
        balls[t][3]=-balls[t][3];
    } else if (balls[t][1]>gameSize.getHeight()-balls[t][4]){
        balls[t][1]=gameSize.getHeight()-balls[t][4];
        balls[t][3]=-balls[t][3];
    }
}
}

public void render(Graphics2D g) {
    // Clear back ground
    g.setColor(Color.black);
    g.fillRect(0,0,gameSize.getWidth(),gameSize.getHeight());

    // Draw balls
    for (int t=0;t<balls.length;t++){
        g.setColor(ballsColor[t]);
        g.fillOval((int)balls[t][0],
                 (int)balls[t][1],
                 (int)balls[t][4],
                 (int)balls[t][4]);
    }
}

public boolean acceptKeystroke(KeyEvent stroke) {
    gameController.stop();
    return true;
}

public void mouseClicked(int x, int y, int button) {
}

public void mouseDoubleClicked(int x, int y, int button) {
}

public void mouseMoved(int x, int y) {
}

public void mousePressed(int x, int y, int button) {
}

public void mouseReleased(int x, int y, int button) {
}

public void mouseWheelMoved(int x, int y, int rows) {
}

public static void main(String[] args) {
    // Get Game controller from properties file.
    gameController=GameController.getConfiguredInstance("Tutorial");
    //Set screen settings
    gameController.setFullscreenMode(true);
    gameController.setRequestedScreenSize(new Size(800, 600));

    //Set Screen buffers
    gameController.setRequestedScreenBuffers(3);

    // Set the Frame control parameters

```

```
gameController.setLogicFPS(60);
gameController.setRenderMayExceedLogic(false);

// Set the Game within the controller.
gameController.setGame(new Tutorial());

// Start the game
gameController.start();
}
}
```