**K. K. WAGH POLYTECHNIC, NASHIK**

HIRABAI HARIDAS VIDYANAGARI, AMRUTDHAM, PANCHAVATI, NASHIK-422003

# DEPARTMENT OF COMPUTER TECHNOLOGY



**K. K. Wagh Polytechnic, Nashik**

**Micro-Project Report**

**Institute Code:** 0078 **Academic Year:** 2022-23 **Program:** Computer Technology

**Course:** Advanced Java Programming **Course Code:** 22517 **Scheme**: I

**Semester**: Fifth **Class:** TYCM – II **Date of Report**: 26 /11 /2021

**Title of Micro-Project:** Developing a Brick Breaker Game

1. **Rationale:**

Advanced Java programming provides us with the hands-on experience on GUI technologies i.e., AWT and Swing, event handling mechanisms and network programming. In this project, various swing components, Event handling methods, etc are used to develop the Brick Breaker Game.

**2.0 Aim /Benefits of Micro-Project:**

The aim of the project is to effectively develop a stand-alone application using advanced concepts of java. The objective of project is to develop Brick Breaker game using Event handling and different Swing components

**3.0 Course Outcomes achieved (COs):**

**CO503.1:** Understand use of Advanced Java Programming Tools.

**CO503.2:** Study Different types of Events and their Handling.

**4.0 Literature Review:**

Brick Breaker (platformer) is a Breakout clone which the player must smash a wall of bricks by deflecting a bouncing ball with a paddle. The paddle may move horizontally and is controlled with left and right arrow keys. When the bricks are destroyed the score of the player increases and when the ball bouncing ball goes below the deflecting paddle the player loses. The player wins the game whenever he breaks all the bricks present on the screen.

**5.0 Actual Methodology followed:**

1. Finalizing team members.
2. Discussing about the topic with mentor/guide/subject teacher.
3. Selection of project topic based on research.
4. Distributing different tasks like collecting information, creating proposal etc. between team members.
5. Collecting the information about the topic.
6. Merging all the information and research collected by team members.
7. Creating final document.
8. Creating Program Code for selected topic using Advanced Java Concepts.
9. Checking Requirements of guide regarding Program code and make Changes Accordingly.
10. Creating all the required documentation according to guidelines and merging created project document in report.
11. Submitting the project along with viva.

**6.0 Actual Resources used:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Name of Resource/material** | **Specifications** | **Quantity** | **Remarks** |
| 1 | Computer system | Asus TUF A15,Processor Ryzen 7 4800h, RAM:16 GB, SSD :512 GB | 01 | Available |
| 2 | JDK 1.7 | JDK: 1.7.0 | 01 | Available |
| 3 | VS Code | Version :1.73 | 01 | Available |
| 4 | Other resources | MS-Word | 01 | Available |

**7.0 Output of The Micro-Project**

• **CODE:**

import *javax*.*swing*.*\**;

import *java*.*awt*.*\**;

import *java*.*awt*.*event*.*\**;

*public* *class* Gameplay *extends* JPanel *implements* KeyListener, ActionListener {

    boolean play = false;

    int score = 0;

    int totalbricks = 21;

    Timer;

    int delay = 0;

    int playerX = 310;

    int ballposX = 120;

    int ballposY = 350;

    int ballXdir = -1;

    int ballYdir = -2;

    MapGenerator map;

    Gameplay() {

        map = new MapGenerator(3, 7);

        addKeyListener(this);

        setFocusTraversalKeysEnabled(false);

        setFocusable(true);

        timer = new Timer(delay, this);

        timer.start();

    }

*public* void paint(Graphics g) {

        // *background*

        g.setColor(Color.*black*);

        g.fillRect(1, 1, 692, 592);

        // *draw map*

        map.draw((Graphics2D) g);

        // *scores*

        g.setColor(Color.*white*);

        g.setFont(new Font("serif", Font.*BOLD*, 25));

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

        // *borders*

        g.setColor(Color.*yellow*);

        g.fillRect(0, 0, 3, 592);

        g.fillRect(0, 0, 692, 3);

        g.fillRect(692, 0, 3, 592);

        // *the paddle*

        g.setColor(Color.*green*);

        g.fillRect(playerX, 550, 100, 8);

        // *the ball*

        g.setColor(Color.*yellow*);

        g.fillOval(ballposX, ballposY, 20, 20);

        // *won*

        if (totalbricks <= 0) {

            play = false;

            ballXdir = 0;

            ballYdir = 0;

            g.setColor(Color.*red*);

            g.setFont(new Font("serif", Font.*BOLD*, 30));

            g.drawString("You won....Scores:" + 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....Scores:" + score, 190, 300);

            g.setFont(new Font("serif", Font.*BOLD*, 20));

            g.drawString("Press enter to restart", 230, 350);

            // *repaint();*

        }

        g.dispose();

    }

    @Override

*public* void keyTyped(KeyEvent e) {

        // *TODO Auto-generated method stub*

    }

    @Override

*public* void keyPressed(KeyEvent e) {

        if (e.getKeyCode() == KeyEvent.*VK\_RIGHT*) {

            if (playerX >= 600) {

                playerX = 600;

            } else {

                moveRight();

            }

        }

        if (e.getKeyCode() == KeyEvent.*VK\_LEFT*) {

            if (playerX < 10) {

                playerX = 10;

            } else {

                moveLeft();

            }

        }

        if (e.getKeyCode() == KeyEvent.*VK\_ENTER*) {

            if (!play) {

                play = false;

                score = 0;

                totalbricks = 21;

                delay = 0;

                playerX = 310;

                ballposX = 120;

                ballposY = 350;

                ballXdir = -1;

                ballYdir = -2;

                map = new MapGenerator(3, 7);

            }

        }

    }

    void moveRight() {

        play = true;

        playerX += 20;

    }

    void moveLeft() {

        play = true;

        playerX -= 20;

    }

    @Override

*public* void keyReleased(KeyEvent e) {

    }

    @Override

*public* void actionPerformed(ActionEvent e) {

        timer.start();

        if (play) {

            if (new Rectangle(ballposX, ballposY, 20, 20).intersects(new Rectangle(playerX, 550, 100, 8))) {

                ballYdir = -ballYdir;

            }

            A: 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.*brickHeigtht* + 50;

                        int brickWidth = map.*brickWidth*;

                        int brickHeigtht = map.*brickHeigtht*;

                        Rectangle rect = new Rectangle(brickx, bricky, brickWidth, brickHeigtht);

                        Rectangle ballRect = new Rectangle(ballposX, ballposY, 20, 20);

                        Rectangle brickRect = rect;

                        if (ballRect.intersects(brickRect)) {

                            map.setBrickValue(0, i, j);

                            totalbricks--;

                            score += 5;

                            if (ballposX + 19 <= brickRect.*x* || ballposX + 1 >= brickRect.*x* + brickRect.*width*) {

                                ballXdir = -ballXdir;

                            } else {

                                ballYdir = -ballYdir;

                            }

                            break A;

                        }

                    }

                }

            }

            ballposX += ballXdir;

            ballposY += ballYdir;

            if (ballposX < 0) {

                ballXdir = -ballXdir;

            }

            if (ballposY < 0) {

                ballYdir = -ballYdir;

            }

            if (ballposX > 670) {

                ballXdir = -ballXdir;

            }

        }

        repaint();

    }

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

        JFrame j = new JFrame();

        Gameplay = new Gameplay();

        j.setBounds(10, 10, 700, 600);

        j.setTitle("BreakoutBall");

        j.setResizable(false);

        j.setVisible(true);

        j.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

        j.add(gameplay);

    }

}

*class* MapGenerator {

    int map[][];

    int brickWidth;

    int brickHeigtht;

    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;// *'1'will detect that this particular brick has not been hit by the ball yet*

            }

        }

        brickWidth = 540 / col;

        brickHeigtht = 150 / row;

    }

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.*white*);

                    g.fillRect(j \* brickWidth + 80, i \* brickHeigtht + 50, brickWidth, brickHeigtht);

                    g.setStroke(new BasicStroke(3));

                    g.setColor(Color.*black*);

                    g.drawRect(j \* brickWidth + 80, i \* brickHeigtht + 50, brickWidth, brickHeigtht);

                }

            }

        }

    }

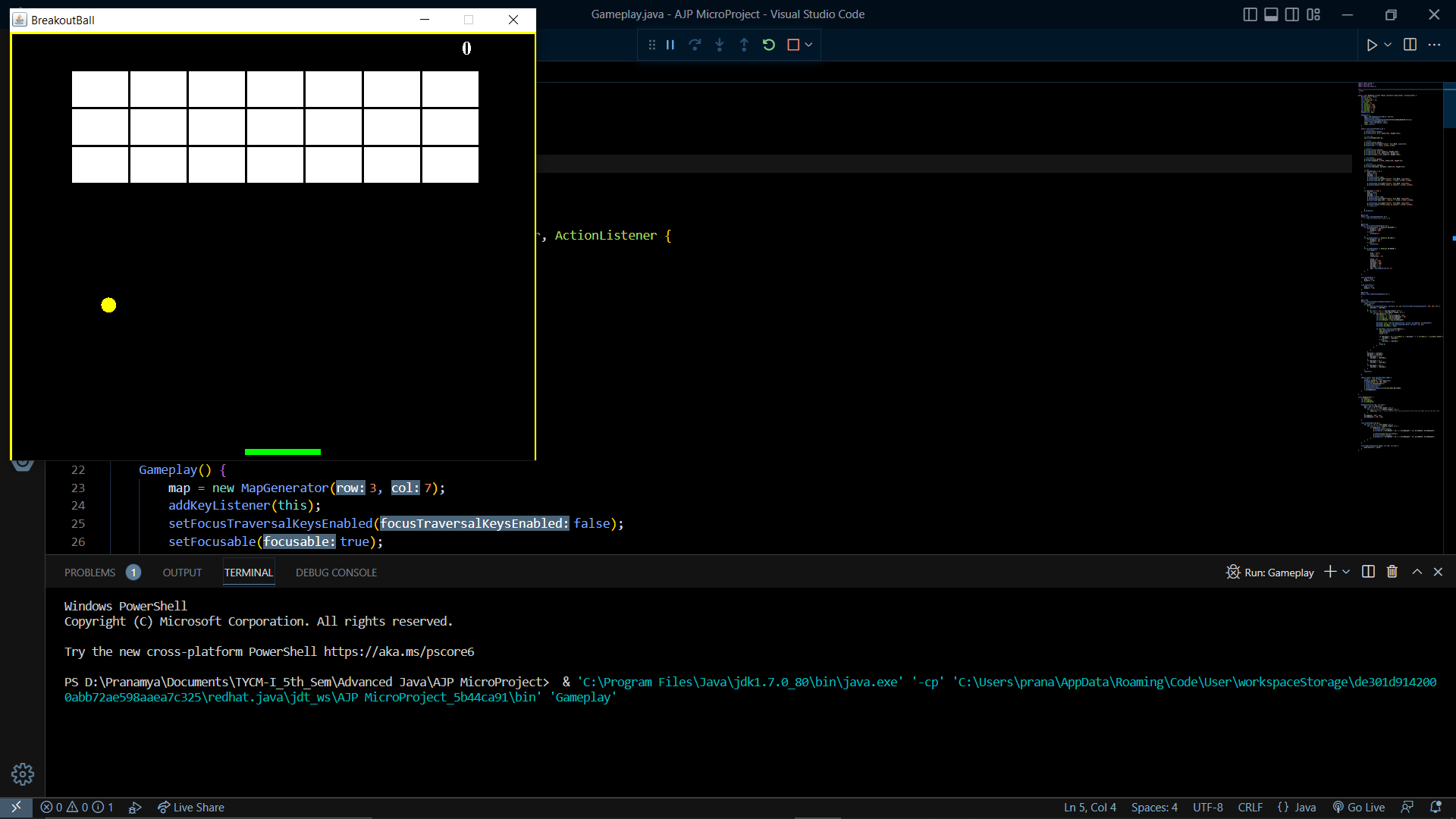
void setBrickValue(int value, int row, int col) {

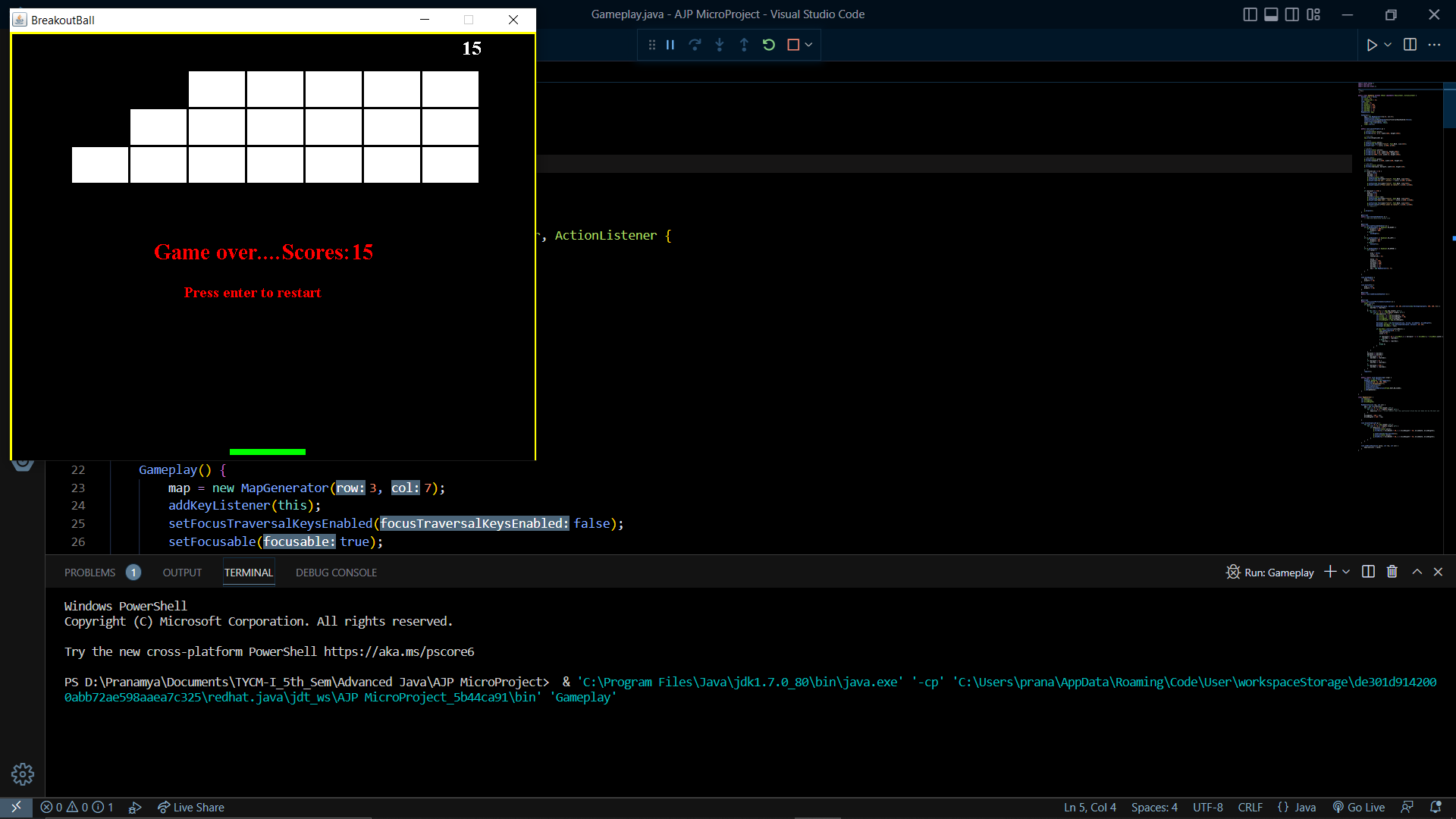
        map[row][col] = value;

    }

}

**Output:**

****

****

**8.0 Skill Developed / Learning outcome of this Micro-Project:**

1. With this project learned different AWT components and Event handling
2. We can effectively use concepts of advanced java programming.
3. Develop standalone application concepts of Java.

**9.0 Applications of Micro Project:**

1. Easy to play
2. Effective and Easy to use.

**10.0 Name of Group Members:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Enrolment | Roll No. | Seat No. | Name of Students | Student Signature |
| 2000780284 | 13 |  | Deshpande Pranamya Nilesh |  |
| 2000780291 | 19 |  | Gaikwad Om Mukund |  |
| 2000780306 | 32 |  | Koli Diptesh Rajkumar |  |
| 2000780317 | 41 |  | Patil Parag Dilip |  |

**Approved by:**

**Name & Signature of Guide: Mr. S.K. Mahajan**

**Date: 16/11/2022**