INTERNSHIP PROJECT

ON

Brick Breaker Game

BACHELOR OF TECHNOLOGY

(COMPUTER SCIENCE & ENGINEERING)

SUBMITTED BY:

Nitesh Kesharwani(190240101067)

UNDER THE GUIDANCE OF:

Mrs. Bhanu Priya

IN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ROORKEE INSTITUTE OF TECHNOLOGY

ROORKEE, UTTRAKHAND, INDIA

(2021-2022)

CERTIFICATE

I hereby certify that the work which is being presented in these entitled “**Brick Breaker Game**” in partial fulfilment of the requirement for the award of degree of Bachelor of Technology and submitted in Department of Computer Science of Roorkee Institute of Technology, Roorkee, is an authentic record of my own work carried out under the supervision of **Mrs Bhanu Priya.**

The matter presented in this report has not been submitted by me anywhere for the award of any other Degree of this or any other institute.

NITESH KESHARWANI

This is to clarify that the above statement made by the candidate is correct to the best of our knowledge.

Date: 16 May 2022

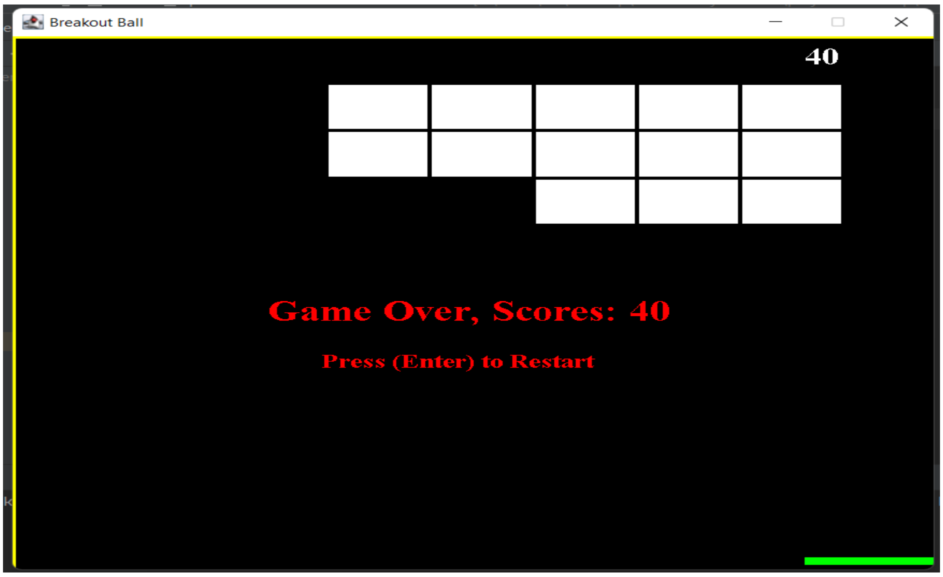
HOD	Project IN charge
(DR. DEEPAK ARYA)	**(Mrs. Bhanu Priya)**

STUDENT INFORMATION

- NAME: ***NITESH KESHARWANI***
- COURSE: ***BTECH***
- BRANCH: ***COMPUTER SCIENCE AND ENGINEERING***
- UNIVERSITY ROLL NUMBER: ***190240101067***
- YEAR: ***3rd YEAR (2019-23)***
- COLLEGE NAME: ***ROORKEE INSTITUTE OF TECHNOLOGY***
- PROJECT NAME: ***BIG BREAKER GAME USING JAVA***
- SUBMITTED TO: ***Mrs. Bhanu Priya***

ABOUT PROJECT:

- Basically this is a brick breaker game (here written in JAVA language) which is designed in such a way that we have a ball and a multiple bricks which will decrease in number whenever the ball will touch the brick and the user will get 5 points for breaking each brick.
- In this project, we use JAVA programming language because JAVA is currently being widely used in developing various games due to the vast pool of library it has, is cross-platform, easy to write, etc.
- Here, we used various JAVA components like:
  - AWT
  - Swing
- This is simple game which can be installed in televisions as well as in smart phones which small kids can play to increase their mind IQ.



Game Over, Scores: 40

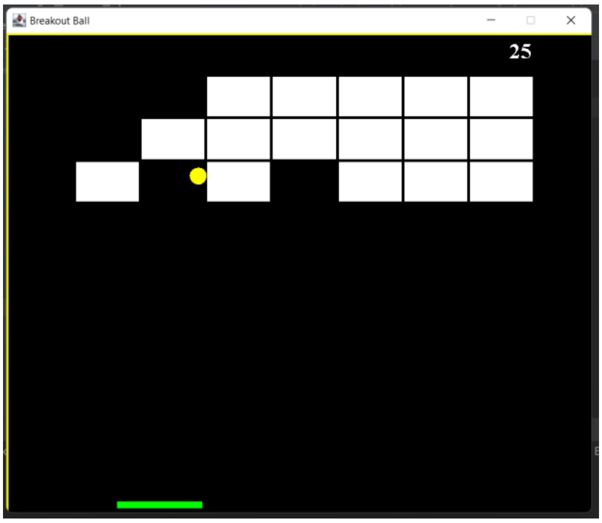
Press (Enter) to Restart

Mechanism involved in developing the Brick Breaker Game using JAVA–

- **Event handling of java**
- **Main Function**
- **Swing component of java**
- **Action Event functions of java**
- **AWT graphics of java**

LANGUAGES USED IN THE PROJECT

In this project, Brick Breaker Game, JAVA programming language is used in both backend and at frontend. The main language used in both is JAVA in which swing, awt, event handling library is also used at great extent.



SOURCE CODE

THE BELOW CODE IS FOR MAIN CLASS:

```
import javax.swing.*;


public class Main {

	public static void main(String[] args) {

		JFrame obj=new JFrame();

		Gameplay gamePlay = new Gameplay();


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

		obj.setTitle("Breakout Ball");

		obj.setResizable(false);

		obj.setVisible(true);


	obj.setDefaultCloseOperation(JFrame.EXIT_ON_C
LOSE);

		obj.add(gamePlay);
```

```
            obj.setVisible(true);

      }

}
```

THE BELOW CODE IS FOR EVENTS OCCURING IN THE GAME AND SETTING OF VARIOUS OBJECTS USED IS THE GAME:

```
import java.util.*;
import java.awt.event.*;


import javax.swing.*;


import java.awt.*;


import javax.swing.*;
import javax.swing.Timer;


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


	private int totalBricks = 48;


	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(4, 12);
      addKeyListener(this);
      setFocusable(true);
      setFocusTraversalKeysEnabled(false);
timer=new Timer(delay,this);
      timer.start();
}

public void paint(Graphics g)
{
      // background
      g.setColor(Color.black);
```

```
g.fillRect(1, 1, 692, 592);

// drawing map
map.draw((Graphics2D) g);

// borders
g.setColor(Color.yellow);
g.fillRect(0, 0, 3, 592);
g.fillRect(0, 0, 692, 3);
g.fillRect(691, 0, 3, 592);

// the scores
g.setColor(Color.white);
g.setFont(new Font("serif",Font.BOLD, 25));
g.drawString(""+score, 590,30);

// the paddle
g.setColor(Color.green);
```

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


  // the ball
  g.setColor(Color.yellow);
  g.fillOval(ballposX, ballposY, 20, 20);


  // when you won the game
  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", 260,300);


g.setColor(Color.RED);
g.setFont(new Font("serif",Font.BOLD, 20));
```

```
          g.drawString("Press (Enter) to Restart",
230,350);
            }


            // when you lose the game
            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.setColor(Color.RED);
          g.setFont(new Font("serif",Font.BOLD, 20));
          g.drawString("Press (Enter) to Restart",
230,350);
```

```
}

    g.dispose();
}

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 = true;
                      ballposX = 120;
```

```
            ballposY = 350;

            ballXdir = -1;

            ballYdir = -2;

            playerX = 310;

            score = 0;

            totalBricks = 21;

            map = new MapGenerator(3, 7);


            repaint();

        }

}

}


public void keyReleased(KeyEvent e) {}

public void keyTyped(KeyEvent e) {}


public void moveRight()

{
        play = true;
        playerX+=20;
    }

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

    public void actionPerformed(ActionEvent e)
    {
        timer.start();
        if(play)
        {
            if(new Rectangle(ballposX, ballposY, 20,
20).intersects(new Rectangle(playerX, 550, 30, 8)))
            {
                ballYdir = -ballYdir;
                ballXdir = -2;
            }
            else if(new Rectangle(ballposX, ballposY, 20, 20).intersects(new Rectangle(playerX + 70, 550, 30, 8)))
            {
                ballYdir = -ballYdir;
                ballXdir = ballXdir + 1;
            }
            else if(new Rectangle(ballposX, ballposY, 20, 20).intersects(new Rectangle(playerX + 30, 550, 40, 8)))
            {
                ballYdir = -ballYdir;
            }

            // check map collision with the ball

            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)
                    {
                        //scores++;
                        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);
                            score+=5;
                            totalBricks--;


                            // when ball hit right or left of brick
                            if(ballposX + 19 <= brickRect.x || ballposX + 1 >= brickRect.x + brickRect.width)
                            {
                                ballXdir = -ballXdir;
                            }
```

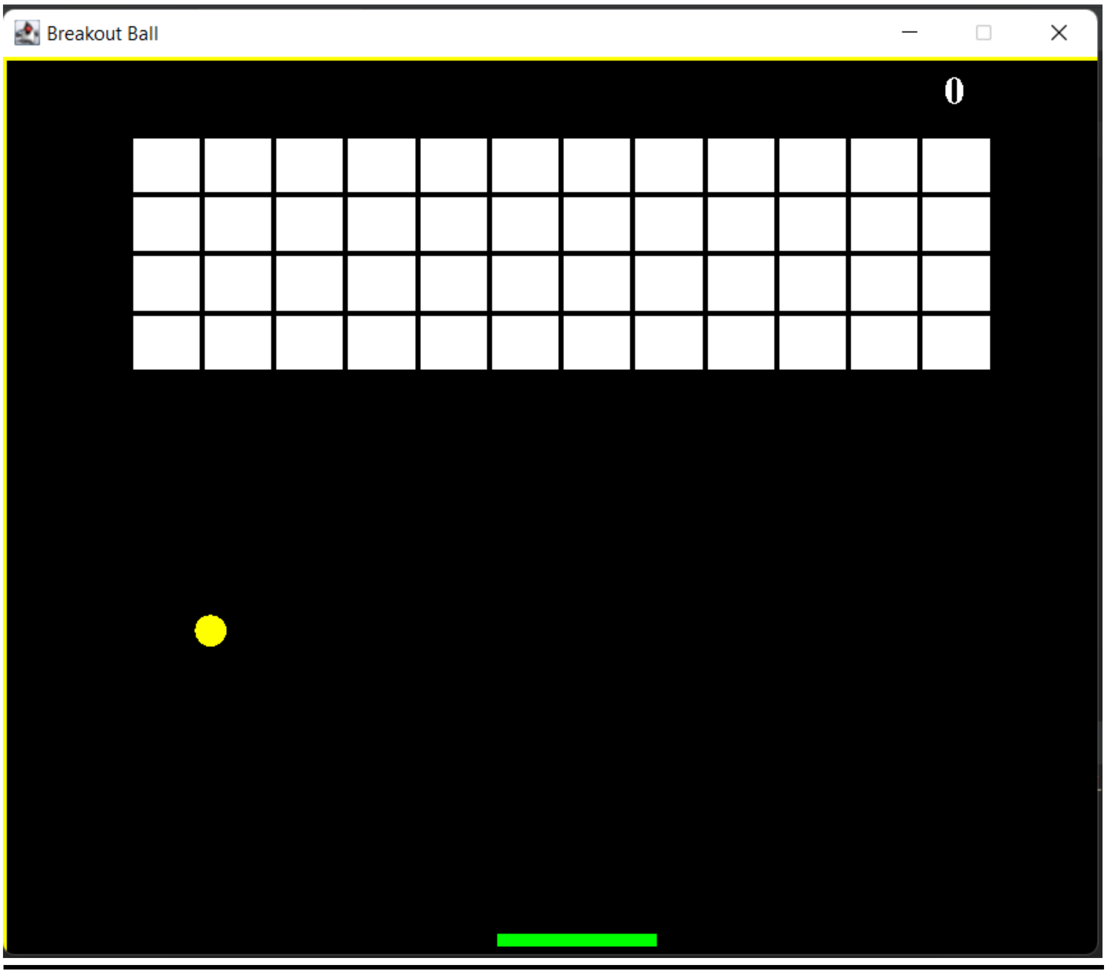
```
                                // when ball hits top or bottom of brick
                                else
                                {
                                    ballYdir = -ballYdir;

                                }

                                break A;
                            }
                        }
                    }
                }

                ballposX += ballXdir;
                ballposY += ballYdir;

                if(ballposX < 0)
                {
```

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

            repaint();
        }
    }
}
```

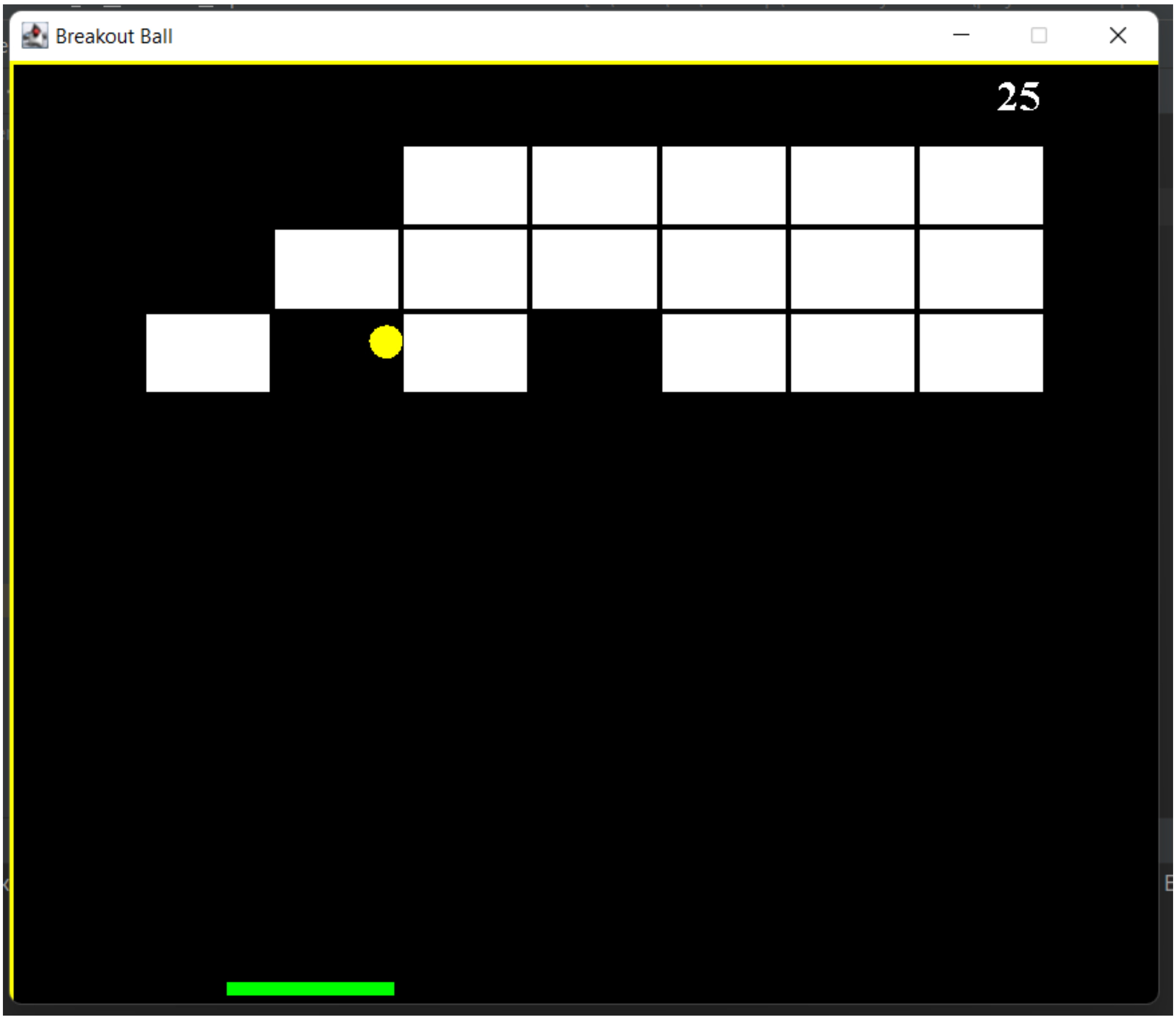
THE BELOW CODE IS FOR VARIOUS GRAPHICS INVOLVED IN THE GAME:

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

                    // this is just to show separate brick, game can still run without it
                    g.setStroke(new BasicStroke(3));
                    g.setColor(Color.black);
                    g.drawRect(j * brickWidth + 80, i * brickHeight + 50, brickWidth, brickHeight);

                }
            }
        }
    }

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

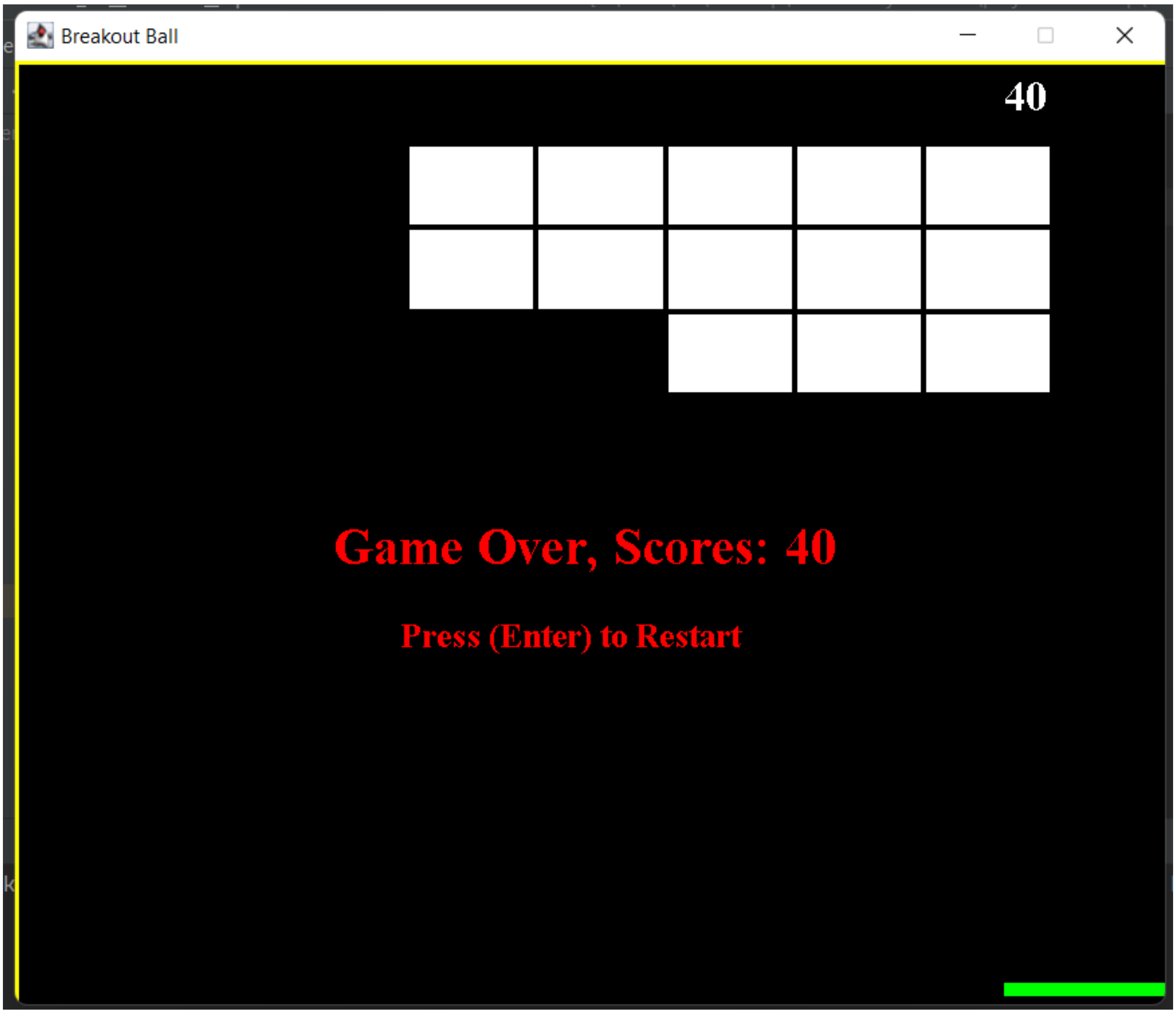
VARIOUS OUTPUTS FROM THE CODE



This is how the game looks in the beginning.



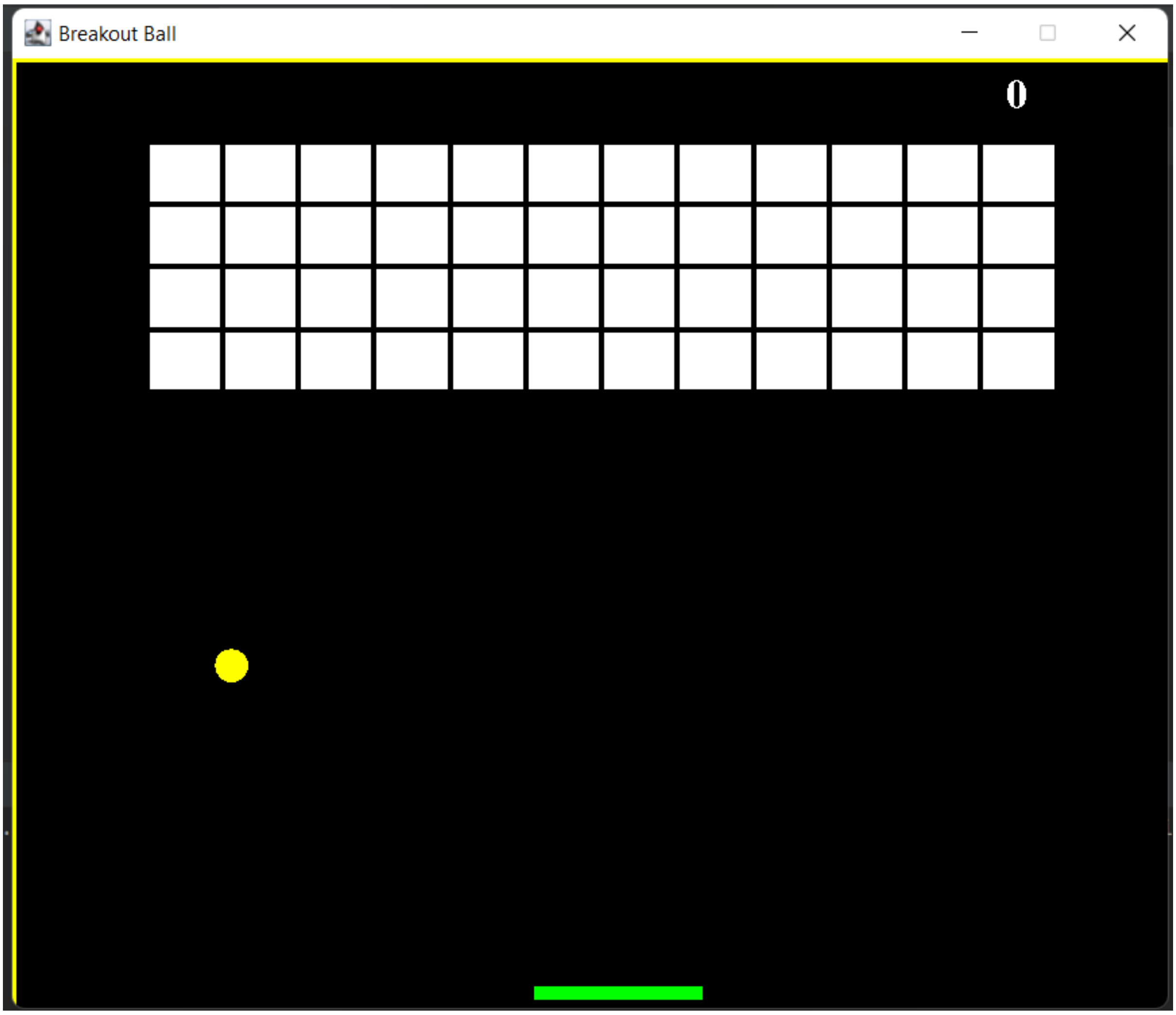
This image is from the middle of the game.



Game Over, Scores: 40

Press (Enter) to Restart

This image is when you not break all the bricks.



This image is when you break all the bricks and win the game.

FUTURE SCOPE OF THE PROJECT

1. This project is quite interesting and has a good future. This game is in market from almost 20 years but still the game holds strong position in market but this game has been made in JAVA programming language which makes this game quite light weighted, faster than before, better user interaction, and so on. This game can still be installed on various multimedia devices and televisions in order to give customers free game. This is very simple game and cheap too and can be easily be circulated in market.

2. The most eye-catching feature of this game is that it is non-addictive and can help small kids to increase their IQ which surely gives it a huge scope in market.

SUMMARY

Brick Breaker game can be used widely because though it is a simple game but still it is highly beneficial for small kids as it will help in increasing their IQ. These games can be pre-installed on Television and Smartphones in order to give users free access to at least one game. We have used various libraries of Java in this project to make it user friendly and interesting for users.