

**INTERNSHIP FINAL PROJECT**

**ON**

**“BRICK BREAKER GAME”**

**IN JAVA**

Submitted by Submitted to

**CHEETI SRUTHI MOTIONCUT**

**SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES Crafting Futures**



**Student’s Declaration**

I CHEETI SRUTHI, a student of B. TECH, Roll No.22TR1A0523 of the Department of CSE of SREE CHAITHANYA INSTITUTE OF TECHNOLOGICAL SCIENCES in Karimnagar, I do hereby declare that I have working on this internship on Java programming in MOTION CUT Crafting Futures.

(Signature and Date)

CH. Sruthi

20/12/2024



**ABSTRACT**

▪ The application “BRICK BREAKER” is an interactive game based upon the classic. The object of brick breaker is to break the bricks that are distributed around the top of the game screen.

▪ The bricks are broken after coming in contact with a ball that bounces around the screen. At the bottom is a paddle that in the classic game moves based on user’s input.

▪ The user has to make sure the ball bounces off the paddle without going off the bottom of the screen.

▪ In our implementation, we use arrow keys to control the paddle’s position, we use to track the position of an actual ping pong paddle held by the user.

▪ The control of the game paddle improves the user experience.

**TABLE OF CONTENTS**

**Page no. s Starts-End**

1.INTRODUCTION……………………………………………………………..1-2

2. SCOPE AND OBJECTIVE………………………………………………….03

3. REQUIREMENTS…………………………………………………………….04

4. TOOLS REQUIRED………………………………………………………….05

5. SETTING UP THE GAME………………………………………………….6-9

6. CODE……………………………………………………………………….10-16

7. SCREENSHOTS…………………………………………………………..17-18

8. RESOURCE AND FURTHER ENHANCEMENTS ……………………...19

9. CONCLUSION………………………………………………………………….20



1. **INTRODUCTION**

Games are a fundamental way that humans interact and learn. They provide so many advantages for people of all interests and abilities. The main benefits of using computer games as learning tools, is related with problem solving, 21stcentury skills, integration of learning and assessment, collaboratives and interactivity, addressing cognitive as well as affective learning issues, and motivation for learning. Gaming is gaining a different level of attention not only from the youngsters but also from different age people. It is creating a virtual world where we can virtually live our life. Gaming has gained importance in Desktop application as well as in Android Application.

▪ This game is a brick ball game that needs to break bricks on the top using one balls. In this game we need to handle one bottom bricks with our arrow keys.

▪ If the game ball touches the bottom area then the game will be over. We should try our best to complete all the levels.

▪ This game will provide the intermediate file to create our own game levels as many as we want. We can challenge our friend by creating complicated game levels.



▪ we chose the game brick breaker because it is already a fun game that we knew we could improve using an actual Ping-Pong paddle.

▪ The actual paddle made the game more interactive and allowed us to take the user’s instinctual movements to allow for a better user experience, in much the same way that current generation video games are employing the technology (e.g. Microsoft Kinect Nintendo Wii, PlayStation Move). Our game was designed to be retro and fun.

▪ It will help to accelerate our practical understanding. This game development project will give a java knowledge with integration and basic animation techniques.



1. **SCOPE AND OBJECTIVE**

As it is a computerbased application game, coding is done in java programming language using Eclipse. to build the game, firstly basic knowledge of java swing and AWT. After requirement of the project were collected and it was decided to build game with different logic flow which can entertain some people.

We already know about the brick breaker game. It has a small ball that hits the bricks with the help of a little platform at the bottom. The player uses this platform to bounce the ball. The more you break the bricks, the more you score. If you miss the ball to bounce the game over. The following are the milestones of the project implementation:

1. Display the bricks and disappear after hitting the ball.
2. Small platform moving left ang right.
3. The ball must bounce after hitting brick and platform at the bottom.
4. Score display.



1. **REQUIREMENTS**
2. **Graphics and GUI:** Implement a visual interface with colourful bricks, a paddle, and a bouncing ball using Java’s graphics libraries.
3. **Game Logic:** Develop core mechanics for paddle movement, ball bouncing, and brick destruction. Include collision detection and a scoring system.
4. **Levels and Difficulty:** Design multiple levels with increasing complexity and adjust difficulty by altering ball speed and introducing challenges.
5. **User Input:** Enable keyboard input for paddle control, and add features like pausing and restarting.
6. **Sound Effects and Music:** Enhance the gaming experience with sound effects for ball hits, brick breaks, and background music.
7. **Game over and Victory Conditions:** Implement conditions for game over and define victory conditions for completing levels.
8. **Optimizations:** Optimize code for better performance, considering features like double buffering.
9. **Documentation:** Provide clear documentation and comments for the codebase to aid understanding.



1. **TOOLS REQUIRED**

Before starting of building Brick Breaker game, make sure we have the following tools and libraries installed:

**Java Development Kit (JDK):** We will need JDK 8 or latest versions to compile and run the Java code.

**Integrated Development Environment (IDE):** Using an IDE like Eclipse, IntelliJ IDEA, or NetBeans is best.



**5.Setting UP The Game**

1. **Create a New java Project**

Open our preferred IDE and create a new java project for our Brick Breaker game.

1. **Create Classes**

Define the necessary classes for the game, such as Game, Ball, paddle, and Brick. These classes will encapsulate the attributes and behaviours of game elements.

**Game.java**

|  |
| --- |
| public class game {  //initialize game elements, set up  the game loop, handle input, and  draw the game.  } |

**Ball.java**

|  |
| --- |
| public class Ball {  //Define the ball’s position, speed, and behaviour, including collision detection with the paddle and bricks.  } |



**Paddle.java**

|  |
| --- |
| public class Paddle {  //Control the paddle’s position and movement.  } |

**Brick.java**

|  |
| --- |
| public class Brick {  //Create and manage bricks, including collision detection with the ball.  } |

1. **Set Up the Game Loop:**

In the game class, create a game loop using the java x. swing library to update the game state, handle user input, and redraw the screen at a constant frame rate.

**Game.java**

|  |
| --- |
| public class Game {  //Initialize the game  public Game () {  //Set up the game window, load resources, and create game objects.  }  //main game loop  public void game Loop () {  while (true) {  //Update game state  Update ();  //Handle user input  Handle Input ();  //Draw the game  Render ();  }  }  //Other methods  } |

1. **Implement Collision Detection**

In the Ball and Brick classes, implement collision detection logic. When the ball collides with a brick, remove the brick and change the ball’s direction. Similarly, when the ball hits the paddle or the game boundaries, update its direction accordingly.

1. **Handle User Input**

Use the java awt event package to handle user input. Capture keyboard or mouse events to move the paddle or perform other in-game actions.

**Game.java**

|  |
| --- |
| public void handle input() {  //handle user input, e.g., move the paddle left or right.  } |



1. **Display the game**

In the game class, use java’s Graphics2D to draw the game elements on the screen. Create a J Panel or a similar component to display game graphics.

**Game.java**

|  |
| --- |
| public void render () {  //Draw the game elements by using Graphics2D.  } |

1. **Add Game Over Logic**

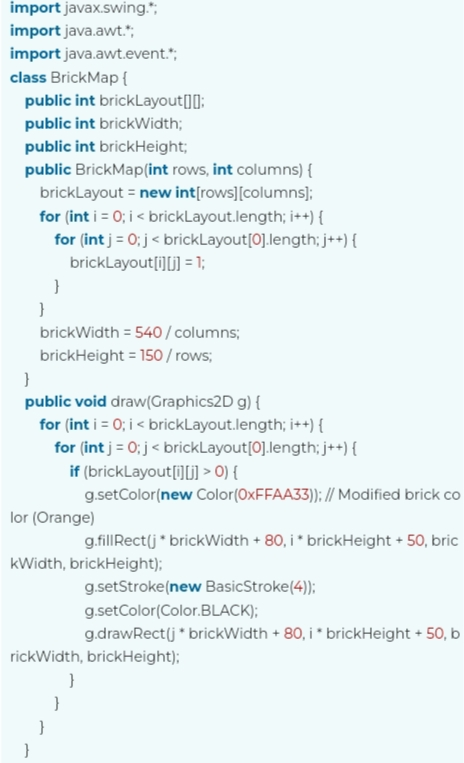
Implement game over conditions, such as when the player loses all lives or breaks all the breaks. Add scoring, lives, and levels to make the game more engaging.

1. **Test the Game**

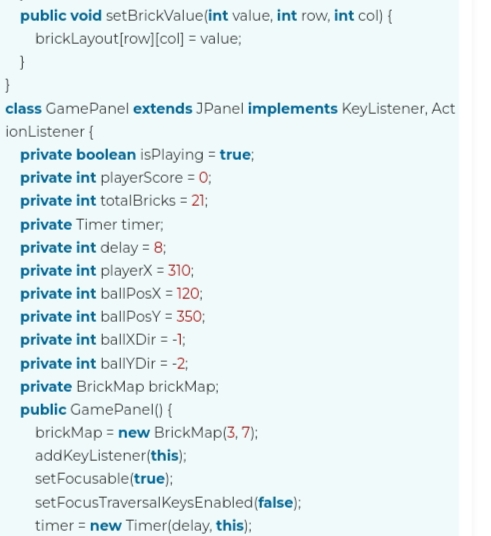
Run the game and test it to ensure everything works as expected. Finetune the gameplay and adjust parameters like ball speed and paddle size for an enjoyable experience.



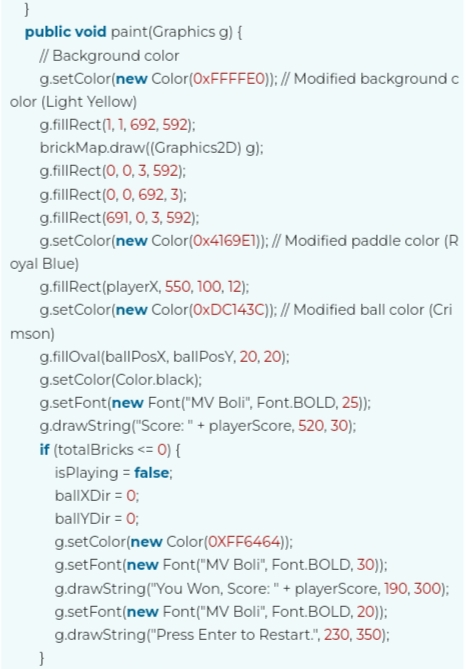
**6.CODE**



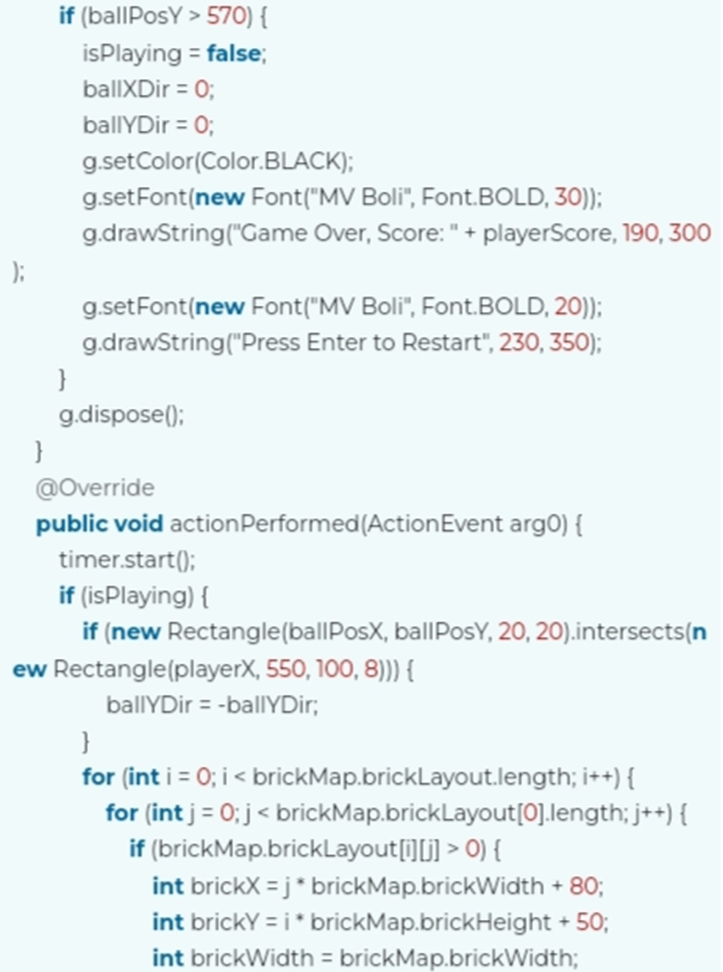


****







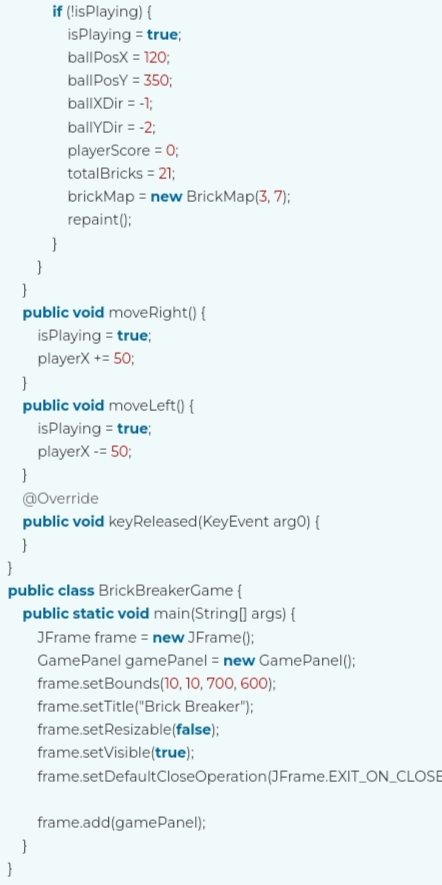






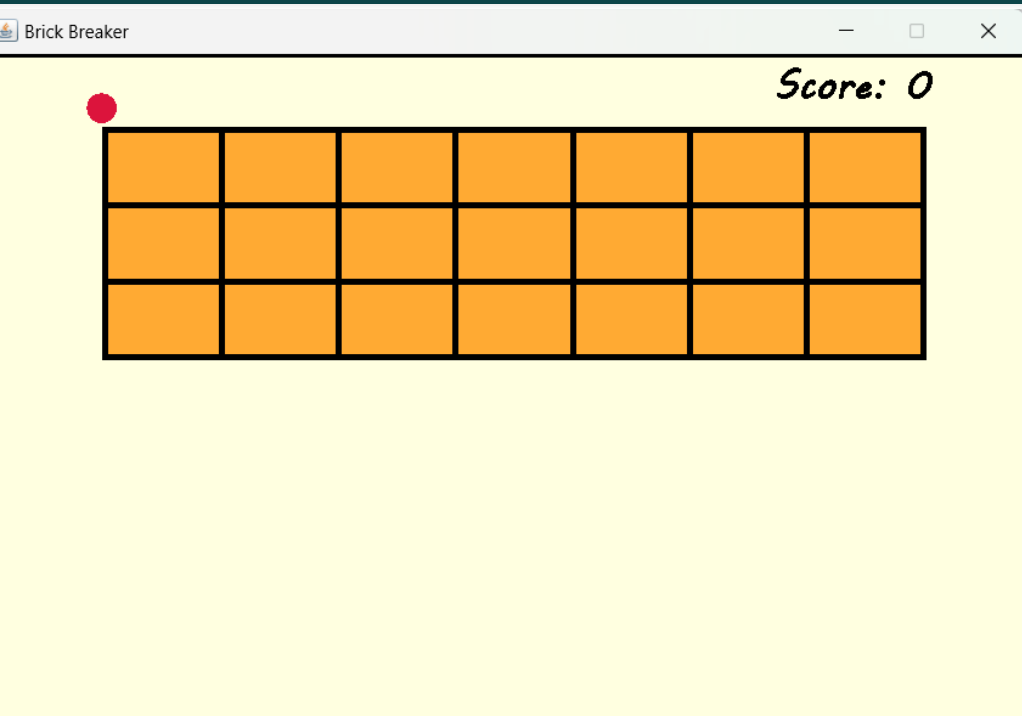




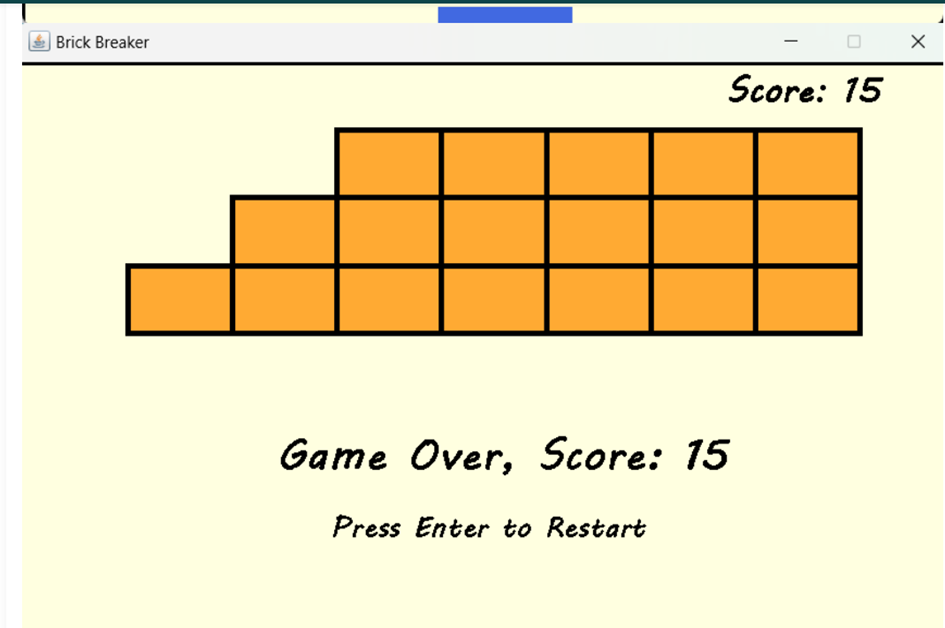




**7.SREENSHOTS**









**8.Resourcse and further Enhancements**

Once we have a working game, consider enhancements such as:

▪ Adding power-ups and different brick types.

▪ Improving graphics and animations.

▪ Implementing sound effects and background music.

▪ Designing multiple levels with increasing difficulty.

▪ Optimizing the game for performance and responsiveness.



**9.CONCLUSION**

This game is based on Java that is an object-oriented programming language. It is a general-purpose programming language, mainly designed to run developed java code on all platforms that support Java without recompilation.

As we know, Java is one of the most popular and in-demand programming languages to learn and it was one of the first language to standardize high-level threading utilities.

Java project is a must for aspiring developers. This project helps developers develop real-world projects to hone their skills and materialise their theoretical knowledge into practical experience. Java has significant advantages both as a commercial language and also a teaching language. Java project provides rigorous compile-time error checking typically associated with Pascal, allowing instructors to introduce students to GUI programming, networking, threads, and other important concepts used in modern-day software. Overall, the java project gives a complete design for the extend language.

Hence here, the main aim of this project was to create a game that would help peoples to entertained themselves. I believe that my project will prove to be a very entertaining game. Building a Brick Breaker game in java is a fun and educational project for aspiring game developers. It covers essential game development concepts and provides a foundation for more complex game projects in the future.

**THANKING YOU!**