2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI) | 979-8-3315-2169-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/IATMSI64286.2025.10985252

Creating an Interactive Breakout Ball Game

Rachna Rathore
Dept.of Mathematics
Jiwaji University
Gwalior, India
rachnarathore1109@gmail.com
https://orcid.org/0000-0001-8753-888X

Saurabh Pratap Singh Rathore
Dept. of Management & Research
ICAPSR
New Delhi, India
rathoresaurabhsingh@gmail.com
https://orcid.org/0000-0002-8390-7569

Manish Kumar
Dept. of CSE
Lovely Professional University
Punjab, India manish.13124@lpu.co.in
https://orcid.org/0009-0004-2528-1600

T. Dhinesh
Dept. of Civil Engineering
Dhanalakshmi Srinivasan College of Eng.
Coimbatore, India

Divyansh Mandhotra
Dept. of CSE (AIML), CSE-APEX
Chandigarh University
Chandigarh, India
divyansh@thecmatrix.com

S. Kaliappan
Division of Research and Development
Lovely Professional University
Punjab India
srini.kal_lpu@yahoo.com

***Abstract*—This research work studies the creation of the traditional Breakout Ball computer game with Java development. In the game, a player paddle at the base of the screen is utilized to deflect a bouncing ball and destroy a wall of blocks at the top of the screen. An object-oriented model is taken, with Java classes representing key entities such as a ball, paddle, blocks, and board. Java Swing is utilized for creating a graphical user interface for the game. Efficient algorithms implement key gaming actions, such as moving a ball, checking collisions, and scoring. Real-time processing and multithreading techniques enable real-time processing and efficient processing of user actions, respectively. Performance analysis factors such as frame rate and CPU consumption in terms of performance and efficiency, respectively, with comparisons with current Breakout Ball implementations, evaluate the performance of the development. Java's aptitude in creating graphical, interactive programs with high-performance and real-time processing capabilities is emphasized through Java's performance in creating Breakout Ball, and Java's aptitude in real-time processing and efficient processing techniques is displayed through Java's performance in Breakout Ball development, respectively.**

***Index Terms*—Breakout Ball, Java, Game Development, Object-Oriented Design, Collision Detection, Multithreading, Performance Evaluation**

I. Introduction

Problem Definition

Despite the overall popularity and ease of Breakout ball, not much work is performed in investigating its contribution towards cognitive and motor skills of its player [1]. Apart from that, its contribution towards difficulty level and skill development in a player is not yet determined [2]. In an endeavor to bridge in such information, in this work, an analysis will be performed in terms of contribution of Breakout ball gaming towards player interest, cognitive, and motor skills at various difficulty levels [3]. One issue that most encounter in Breakout gaming is having a problem in precisely controlling the paddle, and in its failure, one ends up getting frustrated and losing out [4]. Apart from that, Breakout implementations have restrictions in terms of its design, and thus its contribution at various difficulty levels must be determined.

Problem Overview

Breakout is a classic arcade game in which a player controls a paddle to deflect a moving ball and break a group of bricks. Breakout, even with its age, still holds room for examination of its player-player and player-environment relations, and its overall mechanics in general. In a quest for an examination of important factors in successful gaming, such as a moving ball's velocity, a group of bricks' arrangement, and manipulation of a paddle, this work aims to extend player and cognitive skill development, and its overall educational tools, through an extended examination of such factors.

Hardware Specification

The game requires a computer with a minimum spec of

- Processor: Intel Core i3, equivalent to an AMD
- RAM: 2 GB
- Graphics: Integrated, discrete, or shared graphics with a minimum of 512 MB of VRAM
- Storage: 100 MB of free space
- Input Devices: Mouse, keyboard, or a gamepad

Software Specifications

The software setup includes:

- Operating System: windows 7, 8, 10, or mac OS
- Graphics API: DirectX 11 and newer, OpenGL 3.3 and newer
- Development Platform: NetBeans IDE
- Programming Language: Java, with supporting gaming engine

Although Breakout is not a modern-game feature, its mechanism still holds present value for analysis and examination.

II. LITERATURE SURVEY

Rathore et al. (2024) proposed edge and fog computing in smart transportation networks for efficient routing and real-time processing in urban environments [5]. Tural-Polat (2017) developed a general-purpose embedded system for the classic Breakout game, providing a model for optimizing games via settings specific to hardware [6]. Singh and KR (2018) modernized the conventional brick-breaker arcade game with new controls, new levels, and graphical enhancements, but retained its nostalgic value and increased activity for users [7]. Yllana-Prieto et al. (2023) considered escape rooms and breakout games in terms of employing them for delivering subjects in STEM to first-graders, with a view towards developing critical thinking and problem-solving capabilities in future instructors [8]. In a work, Rathore et al. (2024) have used AI for analysis of buyers' feelings, utilizing information-intensive information for a deeper understanding of marketplace trends and buyers' desires [9].

Lankes et al. (2019) analyzed the use of unobtrusive player guidance through gazes in explore games with their "gEYEded" system and demonstrated player gazes' use for added interactivity and challenge [10]. Likewise, Chen et al. (2020) analyzed the use of reinforcement learning in gaming, with Breakout taken as an example, and displayed AI capabilities in gaming improvement and adaptability [11]. Maskeliūnas et al. (2020) designed a mobile educational JavaScript programming game in an ecologic city, utilizing serious gaming methodologies for educational and ecologic aims [12]. Rathore (2023) analyzed AI use in recruitment and selection processes, with AI automation and gamification proving efficient in decision-making and improving candidates' experiences [13]. Nodas et al. (2021) discussed integration of role-playing game (RPG) elements in Breakout games using the platform of LibGDX, depicting a new form in which traditional processes of a game can be creatively blended in a manner to maximize player enjoyment [14]. Shrivastava and Rathore (2024) discussed queueing theory, with a specific case study of Markovian single-server queues with vacation and renege options, and proposed a mathematically implementable model for dealing with changing systems, with real-time gaming and server management a feasible field of application [15]. Sedig et al. (2017) discussed a taxonomic model for describing player–game relations and cognitive gaming, with a discussion of video games' basic processes and its psychological implications [16]. Sanz-Prieto and González (2021) emphasized break-out games' educational gaming potential, citing its use in developing critical thinking and group problem-solving in its players [17]. Rathore et al. (2024) exhibited, in addition, AI application for state-of-the-art analysis of skin cancer in smart environments, depicting AI capabilities in non-conventional gaming environments [18].

III. PROBLEM FORMULATION

State Spaces: The set of all possible configurations that the game might find itself in at any point in time. This could include variables like, but not limited to, position and velocity of the ball, position and size of the paddle, number and position of bricks still remaining on the screen, and so forth. This state space is dynamic, and in almost every frame of the game's run, variables are added, others updated. Taking into consideration this state space of a game, one can better understand its mechanics and be analyzed in a much finer way. Action Space: The actions available to the player at any time that can be performed form this type of space in a Breakout-style ball game. In the present context of a Breakout ball game, this would include moving the paddle left and right, launching the ball from the paddle, power-ups of bonuses that might be available, etc. It could also turn into continuous or discrete action space depending on its implementation in the game. This means that in a continuous action space, it would give players very fine-grained control over the movement of a paddle and ball trajectory; in a discrete action space, players might receive only a select few discrete actions at a given time. Initial State: The state of the Breakout ball game can be defined as the configuration of the game at its most basic and initial stages. This includes such parameters as the position and velocity of the ball, the position and size of the paddle, and the way bricks are arranged on the gaming board. In this very starting position, the ball shall stay at some certain position above the paddle with zero initial velocity. The paddle usually lies at the lowermost part of the game board and generally lies centrally horizontally. The bricks come in some pattern, normally in grid form where they have some predetermined numbers of rows and columns. The initial state forms the basis of the game, and also has influentially significant bearing on how the general challenge that the game shall involve and how it shall play out. For example, changes in basic state—for example, the number or execution of the bricks—obviously change the tough variation and playability of the game. Objective: Remove all the bricks using the provided paddle and ball. In case of successful removal of all bricks, one will emerge victorious; otherwise, it's game over once it goes down below the screen's bottom side.

IV. OBJECTIVE

The objective of this project is to design and develop a Breakout Ball game using the Java programming language within the NetBeans Integrated Development Environment (IDE). The game will be created using the javax.swing and java.awt libraries. It will feature multiple levels with increasing difficulty and various power-ups to enhance gameplay. The game will include a main menu with the following options:

- Start New Game – Allows the player to begin a fresh game.
- How to Play – Provides gameplay instructions and guidelines.
- Settings – Enables customization of player name, soundtrack, and game controls.

• Exit Game – Closes the application. Additionally, the game will introduce power-ups such as extra lives, fastballs, double

balls, and reduced paddle size to add variety and challenge. The game will consist of four difficulty levels:

Easy Level

- Features a simple design with fewer blocks and a larger paddle.
- Ball speed is slow and, for that matter, easier to control.
- Power-ups are easier to access.
- The objective is to remove all of the bricks without exhausting all lives.

Medium Level

- Involves a more complex arrangement with additional bricks and a shorter paddle.
- The ball is moving at a slow pace
- Power-ups are slightly harder to collect.

Hard Level

- Features intricate bricking and a much shorter paddle.
- The ball accelerates and gets increasingly difficult.
- Power-ups are more challenging to obtain.

Expert Level

- The most developed level with a most complex configuration
- The ball moves at high speed, making control difficult.
- Power-ups are extremely difficult to obtain.

Each level boasts its own soundtracks, which correspond with difficulty and intensity of gameplay.

V. METHODLOGY

Requirement for Breakout Game

The game takes place in a rectangular grid with blocks arranged in a column and row configuration with changing colors, shape, and durability. There is a bouncing ball in the board, bouncing off walls, for instance, blocks and a paddle. It starts with a predefined position and velocity, altering its velocity and direction when it collides with an object. The player manipulates a paddle that can move in a restricted direction to stop the ball from dropping off the board. Successful bouncing of the ball aids in shattering bricks and moving through the level in a game. There is a scoring mechanism for tracking a player's points, and these rise with each destroyed brick. There can be added incentives, including power-ups, to reward additional points or bestow a specific skill when a player is in a critical position. The game involves a variety of several levels, with new and specific forms of bricks and increasingly growing difficulty, with new obstacles added in a progression sequence To ensure an exciting experience, specific requirements state game over events. The player wins when out of lives, fails to pass a level in a certain timeframe, or drops the ball too many times. All these mechanics contribute to an immersive and increasingly challenging gaming session.

Game Logic

A rectangular board for a game must first be initialized with a grid of blocks in terms of rows and columns. There is a position with a velocity for a placed ball and a paddle with a starting position and dimensions. There is a start for a loop for a game, and it will run in a continuous manner for as long as a game continues. It will update a game's logic over and over again. To ensure a continuous and even gaming, a Swing Timer updates the paddle, ball, and board positions whenever the loop runs. The paddle can move left and right according to user input through a key press or a move of the mouse. An event listener is added to monitor key presses and/or a move of the mouse, moving the paddle in its respective direction. The ball acts according to its velocity and collides with items placed in the board of the game. It updates its position according to its current velocity. The system detects collisions between the ball and any items, altering its velocity and direction when colliding with them. Collisions for collisions between the paddle, walls, and balls and walls, paddle, and bricks are checked for. A collision algorithm is used for resolving collisions between balls and paddle, and between balls and bricks. On colliding, direction and velocity of the ball is reversed, imparting proper reaction. For a dynamically and exciting gaming environment, bricks have to break when struck and a player must use a paddle to keep a moving ball in motion. For such a case, collisions with walls, paddles, and balls have to be detected when a collision occurs, and velocity and direction have to be reversed in case of a collision. If the ball collides with a brick, its individual brick must be determined and removed from the board, and its player's score must be updated appropriately. On colliding with a paddle, its direction and velocity must both be changed in terms of the incident direction. First, its point of impact with a paddle must be determined, and with its position in relation to a paddle's center, its new path must then be computed. If the ball drops below the board's boundary, a life will be taken away. As such, both the paddle and the ball will have to go back to their starting positions. The game will have to regularly monitor whether the ball drops below the board; when it happens, a life will have to be removed, and both paddle and ball will have to go to their starting positions. All these mechanics make for an immersive and challenging gaming session, with the player having to tactfully maneuver the paddle and react to the motion of the ball in a manner that keeps the activity ongoing.

Check if player cleared all blocks in a board, and move him to the next level in such a scenario. Game Interface

———————-

At the top, include a title for the game The game title, Breakout, will dominate the top of the screen. In the upper-left, a player's current score will be displayed, indicative of points earned through successful breakage of bricks. In the upper right, a lives counter will appear, representing lives yet to be lost. The central part of the screen will have the board, a calm configuration of a paddle, a ball, and a grid of organized bricks. In a grid, in a variety of rows and columns, each of the numerous bricks will reward one point for a player when it is broken. Positioned at the bottom, a rectangular paddle, in a horizontal motion, will allow player direction and manipulation of a ball. In a curved shape, positioned at the top of a paddle, a moving, kinetic ball will move in and

out of a board, bouncing off walls, balls, and even off its paddle counterpart. If the player runs out of lives, a Game Over message will pop up, with a message of the current score and a chance to start over. On the other hand, when all of the bricks have been cleared, a Next Level message will pop up, with a message of the player's current score and a chance to go to the next level, with a new level of difficulty in the continued gameplay.

Testing and Debug

Testing and Debug: Run the game to test if it's playable, the general way it is expected to play out. This will let you check whether the ball physics, paddle movement, and brick destruction are all in place. Test whether the user interface is user-friendly and all buttons together with menus work properly [21]. Do test the game extensively on various devices and corrosponding OS versions for it to go well with each of them. This summary. Hostname research study has presented the development of a new ball game called Breakout Ball using the Java programming language. The aim of this project was to explore several problems related to game development, which includes game mechanics, user interface, and different programming techniques by making this game. They raise valid questions regarding what challenges and considerations are entailed in the development of games, more so within Java programming. During the development of Breakout Ball, several ideas in programming had to be implemented, such as object-oriented programming, processing of events, and in-game physics. They were able to put Java to good use by having access to robust libraries and frameworks—that is, JavaSwing—for creating a rich user interface with an engaging appearance and responsiveness.

VI. OUTPUT

Fig. 1. Welcome Page

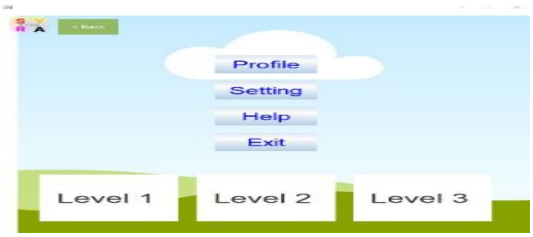

Fig. 2. Setting Page

Welcome page. Main page displays different levels and user profile, setting, help and exit.

Fig. 3. Level One

Fig. 4. Level One

Level 1: contains total 56 bricks with different brick arrangement to make the game more exciting.

Fig. 5. Level Two

Fig. 6. Level Two

Level 2: contains total 84 bricks with different brick arrangement to make the game more exciting.

Game Over: displaying game over with total score after user has finished all lifes . Game Passed : display game passed message with totoal score.

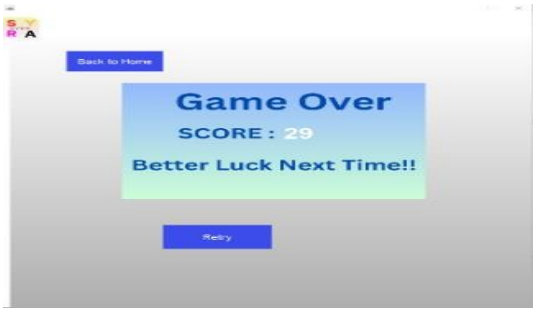

Fig. 7. Result Page

Fig. 8. Result Page

VII. CONCLUSION

The development of an interactive Breakout Ball game demonstrates the effective integration of game design principles with programming techniques to create an engaging user experience. The project successfully implements dynamic gameplay elements, including paddle movement, ball collision detection, and interactive brick destruction. These features provide players with a challenging yet enjoyable gaming environment. Through iterative testing and refinement, the game achieves smooth mechanics and responsive controls, ensuring high playability. The use of modular code design allows for easy scalability, enabling the addition of advanced features like power-ups, levels, and enhanced graphics in future versions. This approach not only ensures a strong foundation for current gameplay but also paves the way for ongoing enhancements. The study highlights the potential of interactive game design to develop both technical skills and creativity. By blending user interaction with programming logic, the game serves as a valuable learning tool for aspiring developers. Overall, this project showcases the versatility of game development as a platform for innovation and entertainment.

REFERENCES

[1] R. Kaźmierczak, R. Skowroński, C. Kowalczyk, and G. Grunwald, "Creating Interactive Scenes in 3D Educational Games: Using Narrative and Technology to Explore History and Culture," Applied Sciences, vol. 14, no. 11. MDPI AG, p. 4795, Jun. 01, 2024.

[2] S. Pathania et al., "An Efficient Electrical-Thermal Co-Design Methodology for Analysis of High-Speed PCB Interconnects," 2023 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO), Winnipeg, MB, Canada, 2023, pp. 154-157.

[3] C.J. Lin, J.Y. Jhang, H.Y. Lin, C.L. Lee, and K.Y. Young, "Using a Reinforcement Q-Learning-Based Deep Neural Network for Playing Video Games," Electronics, vol. 8, no. 10. MDPI AG, p. 1128, Oct. 07, 2019.

[4] M. Verma, A. Kumar and S. Kumar, "Medical Waste Classification using Deep Learning and Convolutional Neural Networks," 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2022, pp. 1-5.

[5] R. Rathore, P. Kaushik, S. S. Sikarwar, H. Joshi, A. K. Mishra and Y. Hudda, "Intelligent Transportation Systems Make Use of Fog and Edge Computing for Navigation," 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2024, pp. 1-6.

[6] S. N. Tural-Polat, "A Complete Embedded System Design of Breakout Game," Journal of Circuits, Systems and Computers, vol. 26, no. 12. World Scientific Pub Co Pte Lt, p. 1750188, Aug. 2017.

[7] Dr. M. Singh and P. KR, "Modernizing the classic brick-breaking arcade game," International Journal of Applied Research, vol. 4, no. 10. AkiNik Publications, pp. 384–388, Jan. 01, 2018.

[8] Yllana-Prieto, D. González-Gómez, and J. S. Jeong, "The escape room and breakout as an aid to learning STEM contents in primary schools: an examination of the development of pre-service teachers in Spain," Education 3-13, vol. 53, no. 1. Informa UK Limited, pp. 15–31, Jan. 03, 2023.

[9] S. P. S. Rathore, J. Patole, G. Tilak, R. Lenka, J. C. Lopez and Priyanka, "Consumer Sentiment Analysis," 2023 International Conference on Smart Devices (ICSD), Dehradun, India, 2024, pp. 1-5.

[10] M. Lankes, A. Haslinger, and C. Wolff, "gEYEded: Subtle and Challenging Gaze-Based Player Guidance in Exploration Games," Multimodal Technologies and Interaction, vol. 3, no. 3. MDPI AG, p. 61, Aug. 22, 2019.

[11] Chen, T. Dewan, M. Trivedi, D. Jiang, A. Aditya, and S. Mohammed, "The Use of Reinforcement Learning in Gaming The Breakout Game Case Study.pdf." Institute of Electrical and Electronics Engineers (IEEE), Apr. 04, 2020.

[12] R. Maskeliūnas, A. Kulikajevas, T. Blažauskas, R. Damaševičius, and J. Swacha, "An Interactive Serious Mobile Game for Supporting the Learning of Programming in JavaScript in the Context of Eco-Friendly City Management," Computers, vol. 9, no. 4. MDPI AG, p. 102, Dec. 17, 2020.

[13] Pratap Singh Rathore, S. (2023). The Impact of AI on Recruitment and Selection Processes: Analysing the role of AI in automating and enhancing recruitment and selection procedures. International Journal for Global Academic & Scientific Research, 2(2), 51–63.

[14] Nodas, D. Tjahjadi, and A. Rahman, "The Implementation of RPG Concept on Breakout Games Using LibGDX Framework," Jurnal Online Informatika, vol. 6, no. 2. Sunan Gunung Djati State Islamic University of Bandung, pp. 233–245, Dec. 26, 2021.

[15] Shrivastava, R. K., & Rathore, R. (2024). Analysis of Single Server Markovian Queueing Model with Differentiated Working Vacation, Vacation Interruption, Soft Failure, Reneging of Customers. International Journal for Global Academic & Scientific Research, 3(3), 01–13.

[16] K. Sedig, P. Parsons, and R. Haworth, "Player–Game Interaction and Cognitive Gameplay: A Taxonomic Framework for the Core Mechanic of Videogames," Informatics, vol. 4, no. 1. MDPI AG, p. 4, Jan. 13, 2017.

[17] M. Sanz-Prieto and G. de Pablo González, "Gamify Gamifying: Learning with Breakouts," Advances in Game-Based Learning. Springer International Publishing, pp. 103–118, 2021.

[18] R. Rathore, H. Yennapusa, S. Kumar, M. Mahawar, S. Miglani and G. Sharma, "Cutting-Edge Skin Cancer Detection in a Smart Ecosystem," 2023 International Conference on Smart Devices (ICSD), Dehradun, India, 2024, pp. 1-5.

[19] Kaushik, P. (2024). Dynamic Data Scaling Techniques for Streaming Machine Learning. International Journal for Global Academic & Scientific Research, 3(1), 1–12.

[20] Monika, A. Taneja, N. Saluja and S. Kumar, "Reconfigurable Antennas for Future Wireless Communication: An Analytical Review," 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2022, pp. 1-6.

[21] Abuhammad et al., "'MedChemVR': A Virtual Reality Game to Enhance Medicinal Chemistry Education," Multimodal Technologies and Interaction, vol. 5, no. 3. MDPI AG, p. 10, Mar. 04, 2021.