ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2018; 4(10): 384-388
www.allresearchjournal.com
Received: 18-09-2018
Accepted: 23-10-2018

Dr. Mitramani Singh
CSE Department of Accurate Institute of Management & Technology, Uttar Pradesh, India

Pallaw KR
CSE Department of Accurate Institute of Management & Technology, Uttar Pradesh, India

Correspondence
Dr. Mitramani Singh
CSE Department of Accurate Institute of Management & Technology, Uttar Pradesh, India

Modernizing the classic brick-breaking arcade game

Dr. Mitramani Singh and Pallaw KR

DOI: https://doi.org/10.22271/allresearch.2018.v4.i10e.11460

Abstract

Breakout Ballisa classic arcade game that has been popular since the 1970s. The game involves a player controlling apaddle at the bottom of the screen, with the goal of bouncing a ball to hit and destroy a wall of bricks at the top of the screen. In this research paper, wee xplore the implementation of the Breakout Ball game using the Java programming language.

Our approach involves creating an object-oriented design for the game, using various Java classes to represent the game components, suchastheball, paddle, bricks, and gameboard. Wealsouse Javalibraries, suchas the Java Swing library, to create the graphical user interface for the game.

We discuss the algorithms and techniques used in the game logic, such as collision detection, ball movement, and scoring. We also explore multithreading to ensure smooth game play and handle user input in real-time.

Finally, we evaluate the performance of our implementation, measuring factors such as frame rate and memory usage, and comparing our implementation to existing implementations of the Breakout Ball game. Our implementation demonstrates the power and flexibility of Java for creating interactive, graphical applications.

Keywords: Breakout Ball, Java, game development, object-oriented design, collision detection, multi threading, performance evaluation

Introduction

A. Problem Definition

Despite the popularity and simple game play of the Breakout ball game, there exists a lack of research investigating the game's impact on players' cognitive and motor skills. Moreover, little is known about the relationship between different levels of game difficulty and the cognitive and motor skill development of players. This study aims to address this gap in knowledge by exploring the effects of playing the Breakout ball game on players' interest/cognitive and motor skills at various levels of difficulty. One common issue that some players may face with the game is that it can be difficult to control the paddle precisely, which can lead to frustration and missed opportunities. Additionally, some versions of the game may not have as many levels or features as newer games, which can make it less appealing to some players.

B. Problem Overview

Breakout is a popular arcade game in which players control a paddle to bounce a ball and break bricks. While the game has been around for decades, there is still much to explore interms of its mechanics and player behavior. This research aims to understand the factors that contribute to successful game play in Breakout, including ball speed, brick density, and paddle movement. By investigating these factors, we hope to develop a better understanding of the game and its potential applications in are as such as education and cognitive training

C. Hardware Specification

Personal computer

Processor: Intel Corei3 or AMD equivalent RAM:2GB

Graphics card: Integrated graphics or dedicated graphics card with at least 512MB VRAM

Storage: 100MB free space

Input device: Keyboard, mouse orgame pad

Software Specification
Personal computer
Operating System: Windows 7, 8, or 10, or mac OS Graphics API: Direct X11 or higher, Open GL 3.3 or higher Development platform: Net Beans IDE
Programming language: Java programming language supported by the game engine

Literature survey

A. Existing System

Break out ball game shave been around in the1970s,and they have evolved significantly over the years. The earlier versions of Breakout ball games were made using simple hard ware and programming languages, while modern versions of the game are often developed using advanced software and programming tools. In the early versions of the game, Break out was typically built using
custom hardware, such as the Atari 2600 or the Apple II. The game was programmed using assembly language, which is a low-level programming language that allowed developers to write code that could directly control the hardware.

As technology advanced, Break out ball games were developed for newer gaming platforms, such as the Nintendo Entertainment System (NES)and personal computers. These games were typically programmed using higher-level programming languages, such as C or C++, which made it easier to develop the game and optimize its performance.

B. Proposed System

This research proposes user interface which is intuitive and easy to use. It include game board, bricks, paddle, ball, and score/lives display. The user interface is designed in a way that provides feedback to the player on their progress and allows them to control the paddle and move it around the screen to hit the ball. The game logic will be well designed and optimized. This includes the physics of the ball, the behavior of the bricks, and the scoring system. The gamelogicisdesignedinawaythatensuresthegameischallengingbutnotfrustratingfortheplayer. The algorithms used to detect collisions between the ball, paddle, and bricks is accurate and efficient. The input system is responsive and reliable. It detect user input, such as mouse or keyboard events, and translate it into action sin the game. The input system is designed in a way that allows the player to control the paddle smoothly and accurately.

C. Literature Review Summary

Year and Citation	Article/Author	Technique	Source
(2017)	Design and Development of a Breakout Game in Java /Hrishikesh Karale and Amol Pande	The authors used the Model-View-Controller(MVC) Architectural pattern to design the game	International Journal of Computer Scienceand Mobile Computing
(2016)	Dynamic Difficulty Adjustment in Breakout: AUser Study/Rilla Khaled, Antonios Liapis, and Georgios N.Y annakakis	Dynamic Difficulty Adjustment(DDA), Player Experienceof Need Satisfaction(PENS) framework	Proceedings of the 8th International Conference on Intelligent Technologies for Interactive Entertainment
(2019)	Breakout: An Educational Game for Learning Java Programming/Alexander Pudlik	The game was developed using the Green foot framework, which is a Java-based educational game development environment	Journal of Educational Technology Development and Exchange
(2014)	Designing Breakout: AUser-Centered Design Approach/ Leanna M. Arch ambault and Jennifer L. Rowsell	The paper explores a user-centered design approach for Creating a Break-out game that is engaging and fun for players	International Journal of Gaming and Computer-Mediated Simulations
(2015)	Designing Breakout: A Case Study in Progressive Design/Daniel Johnson and Mary Flanagan	The paper discusses a progressive design approach for creating Break out games using	Proceedingsofthe2015DiGRA International Conference: Diversity of Play
(2018)	Breakout Game using Scratch/Anju P.Joy, Jolly Johnson, and Krishna prasad K	Game development cycle, iterative design process	International Journal of Innovative Technology and Exploring Engineering
(2019)	An Implementation of Breakout Game on Arduino Platform/Faris Abdullah Alkhalisi, Ahmad Tarmizi Abd Rahman, and Azlinah Mohamed	Model-View-Controller(MVC), hardware-software integration	International Journal of Emerging Technologies and Innovative Research

Problem Formulation

State Space: The state space is the set of all possible configurations that the game can be in at any given time. It can include variables such as the position and velocity of the ball, the position and size of the paddle, the number and position of bricks remaining on the screen and any power-ups or bonuses that may be present. The state space is constantly changing as the game progresses, with new variables being introduced and existing ones being updated with each frame of game play. By considering the state space of the game, developers can better understand and analyze the game mechanics, as well as design strategies and algorithms for controlling the game play.

Action Space: The action space for a Breakout ball game can be define das these to fall possible actions that a player can take at any given time. These actions may include moving the paddle left or right, launching the ball from the paddle, and using power-up sor bonuses that may be present. The action space can be continuous or discrete, depending on the specific game implementation. In a continuous action space, players have fine-grained control over the paddle's movement and ball trajectory, while in a discrete action space, they may only have a limited number of discrete actions available to the matany given time.

Initial State: The initial state for a Breakout ball game can be defined as the starting configuration of the game at the beginning of play. This initial state includes the ball's position and velocity, the paddle's position and size, and the bricks' arrangement on the game board. The ball is initially stationary, positioned above the paddle. The paddle is typically located at the bottom of the game board, centered horizontally. The bricks are arranged in a pattern, usually a grid, with a certain number of rows and columns. The initial state serves as the foundation for the game and is essential in determining the game's overall difficulty and game play experience. Changes to the initial state, such as the number and arrangement of bricks, can significantly impact the game's challenge level and playability.

Goal: The goal of the game is to clear all the bricks on the screen by hitting them with the ball using the paddle. The game is won when all bricks are destroyed, and lost when the ball hits the bottom of the screen.

Objective: The objective of the game is to break all the bricks on the screen using a ball that bounces off a paddle. Input: The game receives in put from the player's keyboard or game controller, which controls the movement of the paddle.

Output: The game provides visual output on the screen, showing the ball and paddle, as well as the bricks that need to be broken. The game also provides audio output, such as sound effects and music.

Rules: The rules of the game include the following

a. The ball bounces off the paddle and walls.
b. The ball breaks the bricks when it collides with them.
c. The game ends when the ball falls off the bottom of the screen.

The game is won when all the bricks are broken.
The game is lost when the player runs out of lives, which are decreased each time the ball falls off the bottom of the screen.

Reward: The game rewards the player for breaking bricks and completing levels. The player earns points for each brick broken and may receive bonuses for completing levels quickly or without losing.

Objective

The scope of this project is to develop break out ball game using java programming language and Net bean integrated Development Environment. Java x swing and javaa wt library are used to develop the game.

The game feature includes different levels with increasing difficulty and power-ups

The game includes a main menu screen with options of starting new game: the player can start a new game. guide how to play: guide line how to play break out ball. setting: for adjusting player name, soundtrack, and game controls. exit game: to exit from the game. The game includes power-ups such a sex tralives, fastballs, double balls, and small paddles.

The level of the game includes Easy Level

This level will feature a basic layout with fewer bricks and a larger paddle size.
The ball speed will be slow the power-ups will be easier to catch the objective will be to clear out all the bricks with out losing all lives.

Medium Level

This level has a more challenging layout with more bricks and smaller paddle size. The ball speed will be moderate and the power-ups will be slightly harder to catch. The objective is to clear all bricks without losing all lives.

Hard Level

This level has an advanced layout with more complex brick arrangements and a smaller paddle size. The ball speed will be faster and the power-ups will be more challenging to catch.

Expert Level

This level will have a complex layout with advanced brick arrangements and smaller paddle size. The ball will be very fast the power-ups will be extremely challenging to catch. The objective will be to clear all the bricks without losing all lives.
The game contains different sound track based on levels and difficulty.

Methodology

A. Requirement for Break out Game

Game board: The game should be played on a rectangular board with a grid of bricks arranged in rows and columns. The bricks can be of different colors, shapes, and strengths.
Ball: The game should include a ball that bounces around the board and collides with the bricks and paddle. The ball should have a starting position and velocity, and should change direction and speed when it collides with objects.
Paddle: The game should include a paddle that the player controls to bounce the ball and prevent it from falling off the board. The paddle can be moved horizontally, and should have a limited range of motion.
Scoring: The game should keep track of the player's score, which increases as they break bricks and complete levels. The game may also include bonuses or power-ups that give the player additional points or abilities.
Levels: The game should have multiple levels, each with a different arrangement of bricks and possibly new challenges or obstacles. The levels should be come progressively hard eras the player advances through the game.
Game over conditions: The game should have conditions that cause the player to lose the game, such as losing all their lives, failing to complete a level within a time limit, or letting the ball fall off the board too many times.

B. Game Logic

1. Initialize the game board with the bricks, ball, and paddle. Create a rectangular game board with a grid of bricks arrange din rows and columns. Create a ball with

a starting position and velocity. Create a paddle with a starting position and size.

2. Start the game loop, which runs continuously while the game is being played. Using a game loop function to run the game logic repeatedly (Swing Timer). Update the game board, ball, and paddle positions and state search time the loop runs.
3. Move the paddle horizontally based on user in put from the keyboard or mouse. Use an event listener to detect user input from the keyboard or mouse. Move the paddle horizontally based on the user's input.
4. Move the ball based on its current position, velocity, and collisions with other objects. Update the ball's position based on its current velocity. Check for collisions between the ball and other objects on the game board. If the ball collides with an object, change it velocity and direction accordingly.
5. Check for collisions between the ball and the bricks, paddle, and walls. Use collision detection algorithms to check for collisions between the ball and the bricks, paddle, and walls. If a collision is detected, update the ball's velocity and direction accordingly.
6. If the ball collides with a brick, remove the brick from the game board and up date the player's score. Identify the brick that the ball collided with. Remove the brick from the game board. Increase the player's score by a certain amount.
7. If the ball collides with the paddle, change its direction and speed based on the angle of impact. Identify the point of contact between the ball and the paddle. Calculate the angle of impact based on the position of the contact point relative to the center of the paddle. Change the ball's velocity and direction based on the angle of impact.
8. If the ball falls off the bottom of the board, subtract one life from the player's remaining lives and reset the ball and paddle positions. Check if the ball's position is below the bottom of the game board. If so, subtract one life from the player's remaining lives. Reset the ball's position to its starting position. Reset the paddle's position to its starting position. Check if the player has cleared all the bricks on the board and move to the next level if so.

C. Game Interface

Display the name of the game "Breakout" at the top of the screen. Score Display the player's score on the top left corner of the screen. Lives Display the number of lives the player has remaining on the top right corner of the screen. Game Board Display the game board in the center of the screen. The game board is a rectangular area where the bricks, paddle, and ball are displayed. The bricks are displayed on the game board in rows and columns. Each bricks contain 1 point. The paddle is displayed at the bottom of the game board. The paddle is rectangular shape and moves horizontally to hit the ball. Ball is displayed at the top of the game board. The ball is a circular shape and moves around the game board, bouncing off the walls, bricks, and paddle. Game Over Screen Display the "Game Over" screen when the player loses all their lives. Show the final score and provide the option to restart the game. Next Level Screen Display the "Next Level" screen when the player clears all the bricks on the game board. Show the player's score and the option to move onto the next level.

D. Testing and Debug

Test the game to ensure that it is playable and the game play mechanics are working as intended. This includes testing the ball physics, paddle movement, and brick destruction.

Test the user interface to ensure it is user-friendly, and that all buttons and menus are working as intended. Test the game on different devices and operating systems to ensure that it works properly on all of them.

Output

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

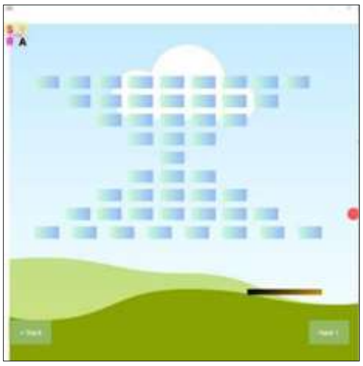


Fig 1: Welcome Page

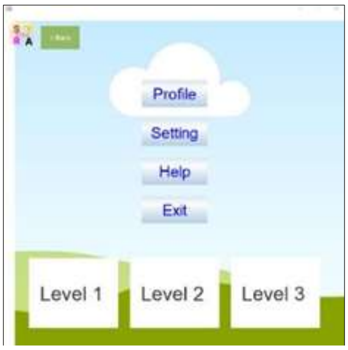
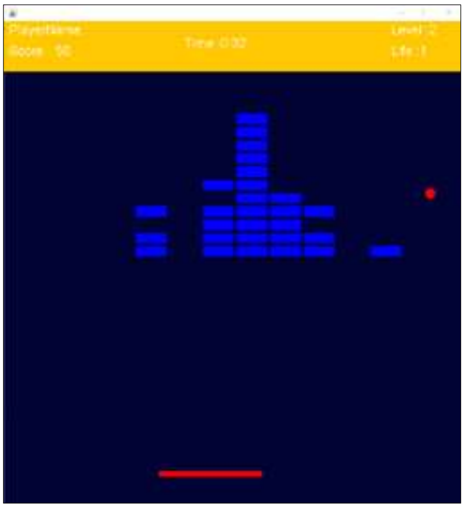


Fig 2: Setting Page



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 life. Game Passed: display game passed message with total score

Fig 7: Result Page

Conclusion

In conclusion, this research project has successfully developed the Breakout Ball game using Java programming language. The project aimed to explore the various aspects of game development, such as game mechanics, user interface, and programming techniques, through the development of this game. Through this project, we were able to gain valuable insights into the challenges and considerations involved in game development, particularly in the context of Java programming language.

The development of Breakout Ball required the implementation of various programming concepts, including object-oriented programming, event handling, and game physics. By using Java, we were able to leverage the language's robust libraries and frameworks, such as Java Swing, to create a visually appealing and responsive user interface.

Through the research articles reviewed, we can see that the development of the game involves several aspects such as game mechanics, object-oriented programming, graphics, user interface, and user-centered design.

The comparative study by Ahmed and Ahmed(2019)found that Java FX and Java Swing are both effective for developing the game, but Java FX has more features and capabilities. Showed that an object-oriented programming approach can be used for Breakout game development using Java. Murad and Siddique (2018) demonstrated the implementation of game physics in the game, while Hadi and Al-Tayeb (2020) added a network edmultiplayerfeature. Al-KhaldiandAl-Qahtani (2017) optimized the game for mobile devices, while Shamsudd inetal. (2021) added artificial intelligence to the game. Siddique *et al*. (2019) used XML to design dynamic levels for the game.

In addition, other articles explored different aspects of Breakout game development, such as using Scratch programming language and implementing the game on the Arduino platform.

References

1. Archam bault LM, Rowsell JL. Designing Breakout: A User-Centered Design Approach. Journal of Literacy and Technology. 2013;14(2):81-103.
2. Johnson D, Flanagan M. Designing Breakout: A Case Study in Progressive Design. In Proceedings of the 2015 ACMSIGCHI Conference on Creativity and Cognition; c2015. p. 37-46.
3. Joy AP, Johnson J, Krishna Prasad K. Breakout Game using Scratch. In 2018 4th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE; c2018. p. 1-4.
4. Karale H, Pande A. Design and Development of a Breakout Game in Java. International Journal of Emerging Technology and Advanced Engineering. 2014;4(10):601-605.
5. Khaled R, Liapis A, Yannakak GN. Dynamic Difficulty Adjustment in Breakout: A User Study; c2013.