Development of a DX Ball Game Using C Programming

Md. Nayeem Ahmed
ECE, NSU
2423510642
nayeem.ahmed.242@nothsouth.edu

Sajid Hasan Abir
ECE, NSU
2422805642
sajid.abir.242@northsouth.edu

Irfan Bin Kamal
ECE, NSU
2423481642
irfan.kamal.242@northsouth.edu

Mir Tausif Siam
ECE, NSU
2422011642
mir.siam.242@northsouth.edu

Abstract

This project involves the development of a text-based DX Ball game using the C programming language. The game demonstrates fundamental programming concepts such as loops, conditionals, functions, and real-time input handling. It simulates the classic arcade gameplay where the player uses a paddle to bounce a ball and break bricks arranged at the top of the screen. The program has features like collision detection, scoring, and life management, providing a simple yet engaging experience.

Introduction

DX Ball, inspired by the classic arcade game Breakout, is a simple, fun, and addictive game. The objective is to control a paddle to bounce a ball upwards to break bricks. This project uses C programming to recreate the game in a text-based environment. It is designed to demonstrate basic programming skills while offering insights into game mechanics like collision detection, structured design, and real-time input processing. The game provides a solid base for integrating more advanced features and enhancements in future iterations.

System Requirements

- **Hardware Requirements**:
  - Processor: Intel Core i3 or higher
  - RAM: 4GB or higher
  - Operating System: Windows/Linux with GCC Compiler
- **Software Requirements**:
  - C Compiler (GCC or Turbo C++)
  - Text Editor (Visual Studio Code, Code::Blocks, etc.)

Proposed System

Objectives

1. To create a text-based DX Ball game using C programming.
2. To implement collision detection, scoring, and life management features.
3. To demonstrate the practical application of basic programming concepts in game development.

Methodology

1. Initialization

- **Bricks**: A 2D array (`bricks`) is initialized to create structured rows of bricks.
- **Ball and Paddle**: The ball and paddle positions are set at the game start.

2. Game Loop

The main program executes an infinite loop comprising three key functions:

1. **`draw()`**: Renders the game state, including bricks, ball, paddle, score, and lives.
2. **`input()`**: Captures player inputs to move the paddle.
3. **`update()`**: Updates ball movement, detects collisions, handles scoring, and manages lives.

3. Collision Detection

- Detects ball collisions with walls, bricks, or the paddle.
- Updates ball direction and score upon collisions.

4. Game Over Handling

- Ends the game when all lives are lost.
- Displays the final score and a message prompting the player to exit.

Implementation Plan

1. **Phase 1: Game Setup**
   - Initialize bricks, ball, and paddle positions.
   - Implement the game board display.
2. **Phase 2: Basic Mechanics**
   - Add ball movement and paddle controls.
   - Detect collisions and update score/lives accordingly.
3. **Phase 3: Testing and Debugging**
   - Test for edge cases, such as ball stuck in a loop.
   - Optimize paddle and ball interactions.
4. **Phase 4: Documentation and Enhancement Suggestions**
   - Prepare project documentation.
   - List possible future improvements (graphics, power-ups, etc.).

Results and Discussion

The DX Ball game was successfully implemented with the following features:

- Real-time input for paddle control.
- Collision detection with walls, bricks, and paddle.
- Scoring system and life management.
- Clear and user-friendly text-based interface.

This project illustrates the effectiveness of structured programming in C to build an interactive application.

Future Scope

1. **Power-Ups**: Add features like multi-ball, lasers, or paddle extensions.
2. **Levels**: Include difficulty progression with varied brick layouts or ball speeds.
3. **Improved Graphics**: Use libraries like SDL or OpenGL for graphical interfaces.
4. **High Scores**: Implement a leaderboard for competitive gameplay.

Conclusion

The DX Ball game is proof to the flexibility of C programming for developing engaging and functional applications. By implementing basic concepts such as loops, arrays, and functions, the project successfully recreates a classic arcade game. It serves as an excellent foundation for exploring advanced programming concepts and game design techniques.

References

1. Kernighan, B. W., & Ritchie, D. M. *The C Programming Language*. Prentice Hall, 1988.
2. Game Design and Development Resources - *GeeksforGeeks*. www.geeksforgeeks.org
3. C Programming Tutorials - *TutorialsPoint*. www.tutorialspoint.com
4. Bug fixes – Stackoverflow. https://stackoverflow.com/