Project Report: Pong Game for FPGA

Yahia Kilany Mahmoud Refaie Seifeldin Elwan
CSCE 2301: Digital Design I
Dr. Dina Mahmoud
Department of Computer Science and Engineering, The American University in Cairo

December 7, 2024

Introduction

The Pong game is a simple yet engaging computer game where players control a paddle to bounce a ball and prevent it from falling off the screen. The game involves interaction between the player and the game environment, requiring reflexes and precision to achieve higher scores. This project serves as an introduction to game development using Verilog, focusing on concepts such as collision detection, graphical rendering, and real-time user input processing.

The objectives of this project are as follows:

- **Game Logic:** Implement the core mechanics of the paddle game, including ball movement, paddle control, and collision detection.
- **Scoring System:** Introduce a scoring mechanism to reward players based on their performance.
- **Graphics and User Interface:** Create an intuitive and visually appealing interface for the game using graphical libraries.
- **Modular Design:** Structure the game code to ensure readability, scalability, and ease of debugging.

This report details the implementation of the Pong game, the modules and algorithms used, the challenges faced during development, and the solutions applied to overcome these challenges.

Modules and Data Structures

The paddle game involves several core modules, each responsible for specific functionality in the game. These modules encapsulate the behavior and attributes of key game elements, ensuring modularity and reusability.

1. Ball Module

Represents the ball in the game, including its position, speed, and direction.

Parameters:

- **Position:** Stores the x and y coordinates of the ball.
- **Velocity:** Represents the speed and direction of the ball as a vector.
- **Radius:** Defines the size of the ball.

Functionality:

- `move()`: Updates the ball's position based on its velocity.
- `checkCollision()`: Detects collisions with the paddle and walls, reversing velocity accordingly.

2. Paddle Module

Represents the player-controlled paddle.

Parameters:

- **Position:** The x-coordinate of the paddle (fixed y-coordinate).
- **Width:** The width of the paddle.
- **Speed:** The speed at which the paddle moves.

Functionality:

- `moveLeft()`: Moves the paddle to the left.
- `moveRight()`: Moves the paddle to the right.
- `checkBounds()`: Ensures the paddle does not move out of screen boundaries.

3. Game Module

Manages the overall game logic, including updates, rendering, and player interaction.

Parameters:

- **Ball:** An instance of the `Ball` module.
- **Paddle:** An instance of the `Paddle` module.

- **Score:** Tracks the player's score.
- **GameState:** Indicates whether the game is running, paused, or over.

Functionality:

- `update()`: Updates the positions of the ball and paddle and checks for collisions.
- `render()`: Draws the game elements on the screen.
- `handleInput()`: Processes user input to control the paddle.

Game Mechanics

The paddle game mechanics revolve around the interaction between the ball, paddle, and screen boundaries. The key algorithms implemented include:

- **Collision Detection:** Uses bounding box algorithms to detect and respond to collisions between the ball and the paddle or walls.
- **Scoring System:** Increments the player's score when the ball successfully bounces off the paddle.
- **Game Over Condition:** Ends the game when the ball falls below the paddle.

Challenges and Solutions

Several challenges were encountered during development:

- **Collision Detection:** Initially, the collision detection for the left paddle was not functioning correctly. The ball would move through the left paddle due to a simple logic error in the collision algorithm.
- **Ball Appearance:** The ball was initially rendered as a square pixel, which detracted from the realism and aesthetic of the game. This was addressed by using a circular shape for the ball, enhancing the overall visual appeal and providing a more authentic gameplay experience.
- **Scoring System Accuracy:** The scoring system initially miscalculated scores, failing to register certain events such as when the ball passed a paddle. The issue was traced to inconsistencies in detecting boundary crossings. By adding precise conditions to check when the ball crossed the scoring zone, the problem was resolved, ensuring accurate scorekeeping throughout the game.

Conclusion

The paddle game project provided a hands-on introduction to game development, emphasizing modular programming, graphical rendering, and real-time interaction. The game mechanics and features were successfully implemented, resulting in an engaging and functional game. Future enhancements could include additional levels, power-ups, and multiplayer support.

Members' Contribution

- **Yahia Kilany:** skladjgskdlagj
- **Seifeldin Elwan:** kgdslajgdsaklgjads
- **Mahmoud Refaie:** gdsjkaglhdsagh