ORGNIZATION OF COMPUTER AND ASSEMBLY LANGUAGE (EL-2003) BRICK BREAKER GAME PROJECT REPORT

Fozan Javaid [23K-0605] Leader

Syed Waleed Hussain [23K-0885]

Abdul Salam [23K-0806]

Brick Breaker Game: Project Report

Team Members

1. Fozan Javaid (23k-0605)

2. Syed Waleed Hussain (23k-0885)

3. Abdul Salam (23k-0806)

1. Introduction

The Brick Breaker game is a classic arcade game implemented in assembly language. It showcases the mechanics of a bouncing ball, paddle control, and breaking bricks, along with scoring and file handling to record user data.

The project is modular, with responsibilities divided among team members for smooth collaboration. The program integrates advanced assembly concepts such as screen rendering, keyboard input, and file operations.

2. Game Features

- Dynamic Ball Movement: The ball moves across the screen, bouncing off walls, the paddle, and bricks.
- Player Interaction: The player can control the paddle's movement using the keyboard.
- Brick Breaking: When the ball collides with a brick, it disappears, and the player earns points.
- User Interface: A welcome screen, prompts for the user's name, and a 'Game Over' screen enhance the user experience.
- Score Saving: User scores are stored in a file (Scores.txt) for future reference.

3. Responsibilities and Tasks

3.1. Fozan Javaid (23k-0605): Ball Movements and Mechanics

Fozan's role focuses on implementing the core mechanics of the ball's behavior, including its movement, collision detection, and interaction with the paddle and bricks.

- Functions:
  - Check_Ball_Current: Manages the ball's position and ensures proper bouncing mechanics.
  - Ball_MOVE: Handles the ball's movement and checks for interactions with the paddle and bricks.
  - Draw_Ball: Renders the ball on the screen at its current position.
- Variables:
  - X_axis_Ball, Y_axis_Ball: Track the ball's position.
  - Acc_X_axis, Acc_Y_axis: Control the ball's acceleration and direction.

3.2. Syed Waleed Hussain (23k-0885): UI, User Input, and Slider Mechanics

Waleed's responsibilities include the design and implementation of the user interface, as well as the mechanics of the slider (paddle) controlled by the player.

- Functions:
    - Draw_Slider: Initializes the slider's position and renders it on the screen.
    - Slider_Move_Left: Allows the player to move the slider left using keyboard input.
    - Slider_Move_Right: Enables the player to move the slider right.
    - printSTART: Displays the welcome screen and prompts the player to enter their name.
- Variables:
    - Slider: Represents the paddle on the screen.
    - X_axis_Slider, Y_axis_Slider: Track the paddle's position.

3.3. Abdul Salam (23k-0806): File Handling and Scoring

Abdul Salam is responsible for managing the scoring system and handling file operations to save user data.

- Functions:
    - StoreInFile: Saves the player's name and score to a file (Scores.txt).
    - ENDUI: Displays the 'Game Over' screen and retrieves the player's name and score.
    - Check_String_Is_Here: Updates the score when the ball collides with a brick.
- Variables:
    - UserName, SCORE: Store the player's name and current score.
    - fileName, Handler: Manage file operations for saving scores.

4. Program Workflow

1. Initialization: The program begins with a welcome screen prompting the player to enter their name.
2. Brick and Slider Setup: Bricks are arranged on the screen, and the slider (paddle) is initialized.
3. Ball Movement: The ball starts moving, and the player can control the slider to prevent it from falling.
4. Brick Collision: The ball breaks bricks upon collision, and the score increases accordingly.
5. Game Over: When the ball misses the slider, the game ends, displaying the player's score and saving it to a file.

5. Technical Details

Development Environment

Assembler: MASM32

Modules:

- Irvine32.inc: Provides essential assembly functions for I/O operations.
- macros.inc: Includes reusable macros for screen rendering and other operations.

Key Concepts

- Screen Rendering: Used to display game elements dynamically.
- Keyboard Input: Captures player input for paddle movement.
- File Handling: Saves player data using assembly-level file operations.

6. Future Enhancements

- Levels: Implement multiple levels with increasing difficulty.
- Enhanced Graphics: Add colors and animations for a better visual experience.
- High Scores: Maintain a leaderboard to track the top players.

7. Conclusion

The Brick Breaker game demonstrates the team's ability to collaboratively design and implement a feature-rich assembly language project. By dividing tasks based on individual expertise, the team successfully combined gameplay mechanics, UI design, and data persistence into a cohesive program.