**Project Report: Brick Breaker Game**

**1. Specification of the Proposed System**

The Brick Breaker game is a classic arcade-style game where the player controls a paddle to bounce a ball and break bricks. The key features of the proposed system include:

- Use of structs and classes to represent game elements such as the paddle, ball, and bricks.

- Use of file handling to store and retrieve the highest score achieved in the game.

- Arrays (vectors) to store collections of bricks for different levels of the game.

**2. Design/Implementation Description**

The software elements that have been implemented to achieve the system include:

- **Paddle and Ball:** Structs representing the paddle and ball, with functions for movement and collision detection.

- **Brick:** Struct representing a brick, with a function to check for collision with the ball.

- **Level Handling:** Two levels are implemented, each with its own set of bricks and background.

- **User Interface:** Graphics are implemented using SFML library, including welcome screens, level transitions, and game over screen.

- **Game Loop:** The main game loop handles user input, updates game elements, and renders the game scene.

- **Sound Effects:** Sound effects are played for various game events such as level transitions and brick breaks.

- **Score Handling:** The game tracks and displays the player's score, updating it when bricks are broken.

**UML Diagram:-**

Player

|

V

[Start Game] --> (Update Game)

|

V

[Move Paddle] --> (Update Paddle Position)

|

V

[Exit Game] --> (Exit Game)

|

V

(Update Game) --> (Check Ball Collision)

|

V

(Check Ball Collision) --> (Brick Broken)

(Check Ball Collision) --> (Ball Out)

|

V

(Brick Broken) --> (Check Win Condition)

|

V

(Check Win Condition) --> (Level Up)

(Level Up) --> (Next Level)

(Next Level) --> (Update Level)

|

V

(Ball Out) --> (Game Over)

|

V

(Game Over) --> (Exit Game)

**3. Test Details**

The following tests could be applied to the program to confirm that it performs according to the specification:

- **Collision Detection:** Test that the ball bounces off the paddle and bricks correctly.

- **Level Transition:** Test that the game transitions to the next level after 60 seconds or when all bricks are broken.

**- Difficulty:** Game difficulty will be increased in level 2. (i.e ball speed will increase and size of paddle will become smaller).

- **Scoring:** Test that the score increments when bricks are broken and displayed correctly.

- **Game Over:** Test that the game ends when the ball falls below the paddle and displays the final score.

Note: Game will be shifted to the next level automatically when the timer of 60 sec will end. And it has only 2 levels for now.

**4. Conclusion**

In conclusion, the Brick Breaker game has been successfully implemented with the specified features. The use of structs, file handling, and arrays has allowed for a modular and extensible design. Further enhancements could include additional levels, power-ups, and improved graphics and sound effects.