User stories

1.Playing a Single Player Game Against a Random Computer

- Given I start the game.
- And I am prompted with a move.
- When I select a move.
- Then it will place my marker (X) on the board.
- Given I have made a move.
- When my turn is over.
- Then the computer will have a random move placed on the board with its marker (O).
- Given a player makes a move.
- And the move results in three consecutive spaces.
- Then the game is over.
- Given a player makes a move.
- And the move is the last empty spot on the board.
- Then the game is over.

Data Models

1.Game Board: The game board can be represented as a 3x3 grid. Each cell in the grid can hold one of three states: empty, X, or O.

2.Players: There are typically two players in Tic Tac Toe, often referred to as Player X and Player O. Each player can be represented by a simple object or data structure containing their name and symbol (X or O).

3.Game State: The game state includes information such as whose turn it is, whether the game is ongoing, and the winner (if any). This can be represented using variables or attributes.

4.Game Logic: Additionally, you'll need functions or methods to handle the game logic, such as checking for a win condition, validating moves, and updating the game state after each move.

Example Game Flow

1. **Start Screen:**
    - Display a start button.
    - Player clicks start to begin the game.
2. **Game Play:**
    - Fruits and bombs fall from the top of the screen at random positions.
    - Player moves the basket left and right to catch fruits.
    - Each fruit caught increases the score.
    - If a bomb is caught, the game ends.
3. **Game Over:**
    - Display the final score.
    - Show a restart button.
    - Player clicks restart to play again.

Technologies

- **HTML5:** Structure of the game.
- **CSS3:** Styling and animations.
- **JavaScript:** Game logic and interactivity.
- **Game Development Library (Optional):** Libraries such as Phaser.js for easier game development and handling animations.