Name : iKumapayi Isaiah Adebanji

Company : Maverick MetaLabs

Test:Unity Game DeveloperCode

Important Note:

The Test follows a top-down approach, utilizing a single MonoBehaviour script to control the overall flow. This design adheres to the Single Responsibility Principle, where each script has a specific role, keeping logic encapsulated and simple.

I introduced the 'UpdateExternal' method and the Initial method, to be called from this single MonoBehaviour script in the Update and start Method respectively, the idea is that I should be able to control the overall game flow from a script that has authority for example the MonsterController contains the information about the Rounds manager, and the monsterManager scripts and it calls the update external methods of this script in its update method.

This approach improves code readability and flexibility, making it easy to understand and modify specific components in our test.

GuiManager:

- Responsibility: Manages UI components in the game.
- Key Features:
    - Displays the current round, total monsters spawned, and the duration of the current round.
    - Provides a pause menu with options to resume, view level stats, and quit the game.
    - Supports additional functionality for displaying saved round data.

Monster:

- Responsibility: Attached to the monster prefab; contains logic for moving the monster.
- Key Features:
    - Handles the movement behavior of individual monsters.
    - Checks if player exit the camera space and then destroys(deactivate)

`CameraSetting`:

- Responsibility: Attached to the Camera; contains logic for setting correctly the spawnPoint of the monsters so it adjust well to the screen
- Key Features:
    - Contains a spawnPointOffset that sets a slight offset to the left of the camera

`MonsterManager`:

- Responsibility: Manages all spawned monsters on the screen.
- Key Features:
    - Tracks active monsters and updates their state.
    - Communicates with the `MonsterController` script for overall game flow.

`MonsterController`:

- Responsibility: Main game controller; the only `MonoBehaviour` script handling updates and starts.
- Key Features:
    - Orchestrates the overall game execution.
    - Manages updates and starts of other scripts in a top-down approach.

`MonsterObjectPool`:

- Responsibility: Handles object pooling for monsters.

- Key Features:
    - Efficiently manages the instantiation and destruction of monsters to improve performance.
    - Ensures that monsters are properly deactivated and returned to the pool.

`RoundManager`:

- Responsibility: Manages game rounds, including spawning monsters and saving round data.
- Key Features:
    - Implements rounds based on the Fibonacci sequence.
    - Manages the duration of each round and the delay before the next round.
    - Saves round data, including round duration and monsters spawned.

Additional Functionality:

- Pause Menu:
    - Pauses the game and provides options to resume, view level stats, and quit.
- UI Elements:
    - Displays the current round, total monsters spawned, and the duration of the current round.
- Level Stats:
    - Allows players to view saved data for each round, including round number, duration, and monsters spawned.

Tests:

- `RoundManagerTest`:
    - Designed to verify the correctness of the CalculateNextFibonacciValue method in the RoundManager class..

Key components of the test class:

IsFibonacci Method:

- A helper method that checks if a given `BigInteger` value is part of the Fibonacci sequence, used bigInteger as opposed to integer value because some of the values where to big

IsPerfectSquare Method:

- Another helper method to check if a given `BigInteger` value is a perfect square.

Sqrt Method:

- A helper method to calculate the square root of a `BigInteger` using binary search.

CalculateNextFibonacciValue_CalculatesCorrectly Test:

- The main test method that iterates through the first 45 rounds of the Fibonacci sequence generated by the `RoundManager`.
- For each round, it calculates the expected Fibonacci value and asserts that it matches the actual value calculated by the `RoundManager`.

Thank you very much for the test, i appreciate it and i am grateful.