Name: Urvashi Gupta Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

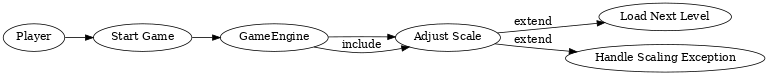
**1. Brief introduction \_\_/3**

The *Scaling Feature* allows the game to adjust difficulty, visuals, and mechanics based on the player’s progress. This ensures players experience smooth difficulty progressing across levels. Scaling includes:

* Increasing enemy strength and count.
* Unlocking new mechanics (e.g., power-ups).
* Adjusting environment complexity.

# Use case diagram with scenario \_\_14

## Use Case Diagrams



## Scenarios

**Name:** Adjust Scale

**Summary:** The System adjusts the game difficulty and mechanics when player progresses to a new level

**Actors:** Player, Game Engine

**Preconditions:** Player has completed the current level. Game engine is running.

**Basic sequence:**

* **Players finish the current level.**
* **System evaluates player performance (time, score, health).**
* **System determines the scaling factor.**
* **Adjust enemies, environment difficulty, and rewards.**
* **Load next level with updated scale.**

**Exceptions:**

* + - * Player finishes with zero score → load easier version of next level.
      * Scaling factor too high → cap adjustments to prevent impossible level.

**Post conditions:** **Player enters next level with adjusted difficulty.**

**Priority:** 1(must have)

**ID:** S01

# Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

## Data Flow Diagrams

A diagram of a company

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.

**Process Descriptions:**

* **Process 4: Enemies** → scaling adjusts health, speed, or damage.
* **Process 5: Boss Fight** → scaling adjusts boss attributes.
* **Process 7: Scaling/Difficulty** → provides adjustments to both enemies and bosses.
* All processes request resources required to function.

WHILE player progresses to next level

IF player performance < threshold

Reduce difficulty (fewer enemies, weaker boss, slower timers)

ELSE IF player performance > threshold

Increase difficulty (stronger enemies, faster pace, more obstacles)

END IF

Cap scaling factor to prevent impossible levels

Apply adjustments to enemy attributes, environment, and rewards

END WHILE

# Acceptance Tests \_\_\_\_\_\_\_\_9

| **Test ID** | **Input (Player Data)** | **Expected Output** | **Notes** |
| --- | --- | --- | --- |
| T01 | Score = 0, Health = Low, Time = Max | Next level loads with reduced difficulty | Boundary: worst performance |
| T02 | Score = Average, Time = Average | Standard scaling applied | Normal case |
| T03 | Score = High, Health = Full | Next level is harder with tougher enemies | Best case |
| T04 | Score = Very High | Scaling capped at maximum difficulty | Prevents impossible levels |
| T05 | Invalid input (negative score) | System ignores invalid value, defaults to 0 | Exception handling |

# Timeline \_\_\_\_\_\_\_\_\_/10 Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Requirements Collection | 2 | - |
| 2. Scaling Algorithm Design | 3 | 1 |
| 3. Level Balancing Rules | 3 | 1 |
| 4. Database Construction | 2 | 2, 3 |
| 5. UI Update | 2 | 4 |
| 6. Programming | 4 | 4 |
| 7. Testing | 3 | 6 |
| 8. Integration with Game build | 2 | 7 |

## Pert diagram

A black background with red arrows with Marfa lights in the background

AI-generated content may be incorrect.

## Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  |
| 2 |  | |  | | 1 | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  |
| 3 |  | |  | | 1 | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  |
| 4 |  | |  | |  | |  | |  | | 2,3 | |  | |  | |  | |  | |  | |  | |  | |  |
| 5 |  | |  | |  | |  | |  | |  | |  | | 4 | |  | |  | |  | |  | |  | |  |
| 6 |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | | 5 | |  | |  | |  |
|  | 0 | 5 | | 10 | | 15 | | 20 | | 25 | | 30 | | 35 | | 40 | | 45 | | 50 | | 55 | | 60 | | 65 | |