**Formato de escenarios y casos de uso  
  
Configuración de los Escenarios**

|  |  |  |
| --- | --- | --- |
| Name | Class | Scenario |
| Setup1 | TransitionPointTest | A transition point is created at (5,5) leading to *RockyMountains*. |
| Setup2 | TransitionPointTest | An invalid transition point is created at (-1,10) with no valid target. |
| Setup3 | PlayerTest | A player exists with name = “Daniel”, health = 3, coordPlayer = (0,0), scenario = *PlainsAndGrasslands*, inventory = empty. |
| Setup4 | PlayerTest | Same player “Daniel” with one Medicine in inventory (name = “bandage”, amount = 1). |
| Setup5 | InventoryTest | Inventory empty with defined capacity. |
| Setup6 | InventoryTest | Inventory completely full at maximum capacity. |
| Setup7 | ObjectiveTest | A valid objective exists with description = “Reach the Rockies”, marked active. |
| Setup8 | ObjectiveTest | An invalid objective exists with no description or status. |
| Setup9 | MedicineTest | A medicine object with valid quantity (5). |
| Setup10 | MedicineTest | A medicine object with invalid quantity (-3). |
| Setup11 | WeaponTest | Revolver with full magazine (capacityAmmo = 6). |
| Setup12 | WeaponTest | Revolver with empty magazine (0/6) but Ammunition object available (10). |
| Setup13 | EnemyTest | An enemy exists with health = 1, coord = (2,2), scenario = *PlainsAndGrasslands*. |
| Setup14 | EnemyTest | Enemy receives damage and HP decreases but not below 0. |
| Setup15 | ScenarioTest | A scenario *PlainsAndGrasslands* where an enemy is added correctly (hp = 10). |
| Setup16 | ScenarioTest | A scenario where adding an invalid enemy (null) is rejected. |
| Setup17 | MapGridTest | A grid is created with valid dimensions height = 20, width = 20. |
| Setup18 | MapGridTest | A grid is created with invalid dimension (height = -5). |
| Setup19 | AchievementTest | Achievement “Explorer” is unlocked and added to BST. |
| Setup20 | AchievementTest | Invalid achievement (null) is not added. |

\* El nombre de los escenarios puede ser setupStage1, setupStage2, etc.

\* La clase es la clase de testing correspondiente al modelo donde acontece el escenario. Por ejemplo si usted está probando User, clase será UserTest.

\* El escenario es la descripción de las condiciones iniciales del escenario.

**Diseño de Casos de Prueba**

**Objetivo de la Prueba:** Verify that transition points are valid or invalid.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| TransitionPoint | validTransitionPointTest(): void | Setup1 | position = (5,5), targetScenario = “RockyMountains” | Transition point is accepted |
| TransitionPoint | invalidTransitionPointTest(): void | Setup2 | position = (-1,10), targetScenario = “” | Invalid transition point → Exception |

**Objetivo de la Prueba:** Verify that the player’s alive status is correctly validated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| Player | positivesIsAliveTest(): boolean | Setup3 | health = 3 | Returns true (player alive) |
| Player | negativesIsAliveTest(): boolean | Setup4 | health = 0 | Returns false (player dead) |

**Objetivo de la Prueba:** Verify that inventory drops items correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| Inventory | PositiveDropItemTest(): void | Setup5 | inventory = [“apple”], drop = “apple” | Item removed successfully |
| Inventory | NegativeDropItemTest(): void | Setup6 | inventory = empty, drop = “bandage” | Nothing removed, inventory unchanged |

**Objetivo de la Prueba:** Verify that objectives are valid or invalid.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| Objective | positiveValidObjectiveTest(): boolean | Setup7 | description = “Reach the Rockies” | Returns true |
| Objective | negativeValidObjectiveTest(): boolean | Setup8 |  |  |

**Objetivo de la Prueba:** Verify that medicine healing quantity is valid or invalid.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| Medicine | validHealingQuantityTest(): int | Setup9 | amount = 5 | Healing accepted |
| Medicine | invalidHealingQuantityTest(): int | Setup10 | amount = -3 | Invalid healing quantity → Exception |

**Objetivo de la Prueba:** Verify that reload time is valid or invalid.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| Weapon | validReloadTimeTest(): int | Setup11 | reloadTime = 3.5 | Reload successful |
| Weapon | invalidReloadTimeTest(): int | Setup12 | reloadTime = -1 | Invalid reload time → Exception |

**Objetivo de la Prueba:** Verify that enemy receives damage correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| Enemy | positiveReceiveDamageTest(): void | Setup13 | hp = 10, damage = 4 | Enemy hp decreases to 6 |
| Enemy | negativeReceiveDamageTest(): void | Setup14 | hp = 0, damage = 5 | HP cannot go below 0 |

**Objetivo de la Prueba:** Verify that enemies can be added to a scenario correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| Scenario | positiveAddEnemyTest(): void | Setup15 | enemy = valid (hp = 10) | Enemy is added to scenario |
| Scenario | negativeAddEnemyTest(): void | Setup16 | enemy = null | Enemy is not added → Exception |

**Objetivo de la Prueba:** Verify that map grid validates height correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| MapGrid | validSetHeightTest(): void | Setup17 | height = 20 | Grid created successfully |
| MapGrid | invalidSetHeightTest(): void | Setup18 | height = -5 | Invalid height → Exception |

**Objetivo de la Prueba:** Verify that achievements are unlocked correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Expected result** |
| Achievement | validIsUnlockedTest(): boolean | Setup19 | achievement = “Explorer” | Returns true, added to BST |
| Achievement | invalidIsUnlockedTest(): boolean | Setup20 | achievement = “” | Returns false |

\* Una prueba se compone de un conjunto de casos de prueba.

\* Cada fila representa un ***caso de prueba*** diferente

\* En el objetivo de la prueba debe escribir una descripción sobre qué es lo que específicamente está probando del modelo del programa.

\* La clase es la clase del modelo que está siendo puesto a prueba.

\* El método es específicamente el método de la clase que está siendo puesto a prueba.

\* El escenario se refiere al nombre del escenario que usted definió. Todos los casos de prueba corresponden a escenarios.

\* Los valores de entrada son valores que entran al método puesto a prueba.

\* El resultado esperado es lo que se espera que suceda luego de ejecutar el método.