**Plan de Pruebas de Proyecto**



**Autor David Bolivar Saldarriaga-Juan Fernando Giraldo Fecha 21/11/2017**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el jugador se mueva de manera efectiva en las posiciones X, para las distintas direcciones |  |  |  |
| **Clase:** | Jugador | **Método:**  testMovimiento() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método reduzca 10 posiciones en X al jugador cuando se dirige hacia la izquierda** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónX = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario) | **dirección = IZQUIERDA** | La posición en X del jugador es:  PosX = 790 |
| **2** | **Se prueba que el método incremente 10 posiciones en X al jugador cuando se dirige hacia la derecha** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónX = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario) | **dirección = DERECHA** | La posición en X del jugador es:  PosX = 810 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el jugador se mueva de manera efectiva en las posiciones X, para las distintas direcciones cuando este saltando |  |  |  |
| **Clase:** | Jugador | **Método:**  testMovimientoEnSalto() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método reduzca 3 posiciones en X al jugador cuando se dirige hacia la izquierda en medio del salto** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónX = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario) | **dirección = IZQUIERDA** | La posición en X del jugador es:  PosX = 797 |
| **2** | **Se prueba que el método incremente 3 posiciones en X al jugador cuando se dirige hacia la derecha en medio del salto** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónX = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario) | **dirección = DERECHA** | La posición en X del jugador es:  PosX = 803 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el jugador incremente sus posiciones en Y y cambie sus imágenes respectivamente a la altura cuando esta saltando |  |  |  |
| **Clase:** | Jugador | **Método:**  testSaltar() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método reduzca 2 posiciones en Y al jugador cuando este saltando y que cambie su imagen a la esperada si no ha avanzado más de 40 posiciones en Y** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónY = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario) |  | La posición en Y del jugador es:  PosY = 798  La imagen del jugador es “jump1I.png” |
| **2** | **Se prueba que el método reduzca 2 posiciones en Y al jugador cuando este saltando y que cambie su imagen a la esperada si ha avanzado entre 40 y 80 posiciones en Y** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónX = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario) | **PosY = 750** | La posición en Y del jugador es:  PosY = 748  La imagen del jugador es “jump2I.png” |
| **3** | **Se prueba que el método reduzca 2 posiciones en Y al jugador cuando este saltando y que cambie su imagen a la esperada si ha avanzado entre 80 y 120 posiciones en Y** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónX = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario) | **PosY = 700** | La posición en Y del jugador es:  PosY = 698  La imagen del jugador es “jump3I.png” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el jugador incremente sus posiciones en Y y cambie sus imágenes respectivamente a la altura cuando esta saltando |  |  |  |
| **Clase:** | Jugador | **Método:**  testGravedadJugador() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método incremente 5 posiciones en Y y su imagen sea la esperada para cuando el jugador este bajando** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónY = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario) |  | La posición en Y del jugador es:  PosY = 805  La imagen del jugador es “jump3I.png” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el jugador pierda vida cuando un proyectil enemigo se encuentre en su posicion |  |  |  |
| **Clase:** | Jugador | **Método:**  testGolpeado() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método reduzca la vida del jugador cuando un proyectil se encuentre en su posición.** | **setUpEscenario2():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónY = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario)  Se crea un proyectil de robot volador con posición 800 tanto para X como para Y y con daño 10 |  | La vida el jugador es:  Vida = 90 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el método es capaz de crear un proyectil y agregarlo a la lista enlazada de proyectiles del escenario |  |  |  |
| **Clase:** | Jugador | **Método:**  testDano() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método agregue un proyectil a la lista enlazada de proyectiles del escenario** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un jugador con posiciónY = 800 que le entra como parámetro el escenario.  jugador = new Jugador(escenario) |  | El proyectil agregado por el jugador a la lista enlazada de proyectiles, es diferente de null.  PrimerProyectil != null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el jugador se mueva de manera efectiva en las posiciones X, para las distintas direcciones |  |  |  |
| **Clase:** | Boss | **Método:**  testMovimiento() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método reduzca 5 posiciones en X al boss pues cambia su posición cuando esté por encima de 700 en X** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un boss con posiciónX = 705 que le entra como parámetro el escenario.  boss = new Boss(escenario) |  | La posición en X del boss es:  PosX = 700 |
| **2** | **Se prueba que el método incremente 5 posiciones en X al boss cuando se dirige hacia la derecha** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un boss con posiciónX = 500,que le entra como parámetro el escenario.  boss = new Boss(escenario) |  | La posición en X del jugador es:  PosX = 810 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el jugador pierda vida cuando un proyectil enemigo se encuentre en su posición |  |  |  |
| **Clase:** | Boss | **Método:**  testGolpeado() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método reduzca la vida del boss cuando un proyectil del jugador se encuentra en su posición.** | **setUpEscenario2():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un boss con vida= 2000, que le entra como parámetro el escenario.  boss = new Boss(escenario)  Además, se crea un proyectil de jugador con posiciones 550 en X y 30 en Y con daño =20. |  | La vida el boss es:  Vida = 1980 |
| **2** | **Se prueba que el método cambie el estado del boss “enAtaque” cuando su vida % 3 == 0** | **setUpEscenario2():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un boss con vida= 2000, que le entra como parámetro el escenario.  boss = new Boss(escenario)  Además, se crea un proyectil de jugador con posiciones 550 en X y 30 en Y con daño =20. | **SetVida(1820)** | La vida el boss es:  Vida = 1800  enAtaque = true |
| **3** | **Se prueba que el método cambie el estado de activo del boss a false cuando su vida sea 0** | **SetUpEscenario2():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un boss con vida= 2000, que le entra como parámetro el escenario.  boss = new Boss(escenario)  Además, se crea un proyectil de jugador con posiciones 550 en X y 30 en Y con daño =20. | **SetVida(0)** | activo = false |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el método es capaz de crear y añadir un proyectil a la lista de proyectilesBoss del escenario |  |  |  |
| **Clase:** | Jugador | **Método:**  testDano() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método agregue un proyectil a la lista de proyectilesBoss del escenario** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un boss que le entra como parámetro el escenario.  boss = new Boss(escenario) |  | El proyectil agregado por el boss a la lista proyectilesBoss, es diferente de null. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el método es capaz de crear y añadir un proyectil a la lista de proyectilesRobotVolador del escenario |  |  |  |
| **Clase:** | RobotVolador | **Método:**  testDano() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método agregue un proyectil a la lista de proyectilesRobotVolador** **del escenario** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un Robot Volador que le entra como parámetro el escenario.  robotV = new RobotVolador(escenario) |  | El proyectil agregado por el boss a la lista proyectilesRobotVolador, es diferente de null. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el método es capaz de reducir la vida del robot volador o volver falso el estado activo si su vida es 0 |  |  |  |
| **Clase:** | RobotVolador | **Método:**  testGolpeado() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método reduzca la vida del robotVolador al estar en contacto con un proyectil** | **setUpEscenario2():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un Robot Volador con vida = 80, que le entra como parámetro el escenario.  robotV = new RobotVolador(escenario)  Además, se crea un proyectil de jugador con posiciones 800 en X y 15 en Y con daño =20. |  | La vida del robotVolador es:  Vida = 60 |
| **2** | **Se prueba que el método cambie el estado activo del robot a false si su vida = 0** | **setUpEscenario2():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un Robot Volador con vida = 80, que le entra como parámetro el escenario.  robotV = new RobotVolador(escenario)  Además, se crea un proyectil de jugador con posiciones 800 en X y 15 en Y con daño =20. | **SetVida(20)** | La vida del robotVolador es:  Vida = 0  activo = false |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el método es capaz de hacer que el robot volador siga al jugador en las posiciones X |  |  |  |
| **Clase:** | RobotVolador | **Método:**  testMovimiento() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el robot volador sigue al jugador con precision** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un Robot que le entra como parámetro el escenario.  robotV = new RobotVolador(escenario) | **Jugador.set**  **Movimiento(true)**  **Jugador.setPosX(792)** | La posicion X del robot volador es:  PosX = 792 |
| **2** | **Se prueba que el robot volador no sigue al jugador mientras este no este en movimiento.** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un Robot que le entra como parámetro el escenario.  robotV = new RobotVolador(escenario) | **Jugador.set**  **Movimiento(false)**  **Jugador.setPosX(784)** | La posicion X del robot volador es:  PosX = 792 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo:** | Se busca probar que el método reduce la vida del misil cuando este está en contacto con un proyectil del jugador |  |  |  |
| **Clase:** | Misil | **Método:**  testGolpeado() |  |  |
| **Caso #** | **Descripción de la prueba** | **Escenario** | **Valores de entrada** | **Resultado** |
| **1** | **Se prueba que el método reduce la vida del misil cuando este se encuentra en la posicion de un proyectil del jugador** | **setUpEscenario1():**  Se crea un escenario nivel 1:  escenario = new Escenario(1)  Se crea un misil  Con posiciones 500 tanto en X como en Y.  Se crea un proyectil con posiciones 621 en X y 666 en Y, con 20 de daño |  | La vida del misil es:  Vida = 20 |