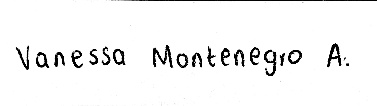
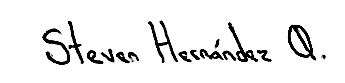
**Informe de responsabilidades (Ejemplo)**

La subida de este documento firmado al repositorio es obligatoria para acceder a la sustentación y debe hacerse el día anterior a la misma. Este documento es obligatorio solamente para equipos de más de 1 integrante.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del componente desarrollado (clase, métodos o funciones)** | **Descripción breve de dicho componente** | **Porcentaje de complejidad respecto al desarrollo global** | **Desarrollador principal (nombre de sólo una persona)** |
| Clase Personaje | Es la clase base abstracta para cualquier entidad viva en el juego, como el Jugador y el Enemigo. Define las propiedades comunes como el sprite, la vida, la posición y los métodos para manejar las animaciones generales de movimiento y mostrar su barra de vida. | 15% | Steven Hernández |
| Clase Jugador | Representa al personajecontrolable por el usuario. También hereda de Personaje y es responsable de manejar la entrada del teclado, las animaciones de movimiento, salto y ataques especiales, la gravedad, y la interacción con bonificaciones y el entorno. | 20% | Vanessa Montenegro |
| Clase Enemigo | Es la clase que define los oponentes del jugador. Hereda de Personaje y añade comportamientos predeterminados como moverse entre plataformas, atacar al jugador (disparar o golpear) y gestionar su propia lógica de vida y muerte. | 15% | Vanessa Montenegro |
| Clase Ataque | Representa los proyectiles o golpes que los personajes pueden lanzar. Se encarga de gestionar su movimiento, ya sea rectilíneo o parabólico (con física de gravedad), y de saber qué personaje lo originó para evitar que se dañe a sí mismo. | 10% | Steven Hernández |
| Clase Obstaculo | Representa elementos estáticos en el escenario con los que los personajes pueden interactuar, como bloques o paredes. Su funcionalidad principal es ser una barrera y, opcionalmente, poder destruirse dejando residuos. | 3% | Steven Hernández |
| Clase Bonificacion | Es la clase para los ítemsrecolectables en el juego. Su función principal es representarse visualmente con un sprite que flota y almacenar un valor de carga que afecta el estado del jugador al ser recogida. | 7% | Steven Hernández |
| Clase MainWindow | Es la ventana principal y el orquestador del Nivel 1 del juego. Se encarga de inicializar la escena, posicionar al Jugador y Enemigo, crear los bloques y muros del laberinto, ubicar las bonificaciones y gestionar la interfaz de usuario de ese nivel, como la barra de súper. | 15% | Steven Hernández |
| Clase MainWindowDos | Esta clase es la ventana y orquestador del Nivel 2. Su función es configurar la escena específica de este nivel, crear las plataformas, posicionar a los personajes y gestionar las barras de vida tanto del jugador como del enemigo, así como la lógica de fin de juego para este nivel. | 15% | Vanessa Montenegro |

**Tabla resumen**

|  |  |  |
| --- | --- | --- |
| **Nombre del integrante**  **(Una fila por integrante del equipo)** | **Nombre de todos los componentes desarrollados** | **Porcentaje total desarrollado**  **(la suma de los ítems de esta columna debe dar 100)** |
| Vanessa Montenegro | Clase Jugador, MainWindow, Enemigo | 50% |
| Steven Hernández | Clase Personaje, Obstaculo, Bonificacion, Ataque | 50% |
|  | Total | 100% |

Vanessa Montenegro: Steven Hernández:

**Nota:** La repartición de responsabilidades especificada en este formato no exime a ninguno de los miembros del equipo de la responsabilidad de conocer y explicar el análisis y diseño de las estrategias que fundamentan la solución entregada.