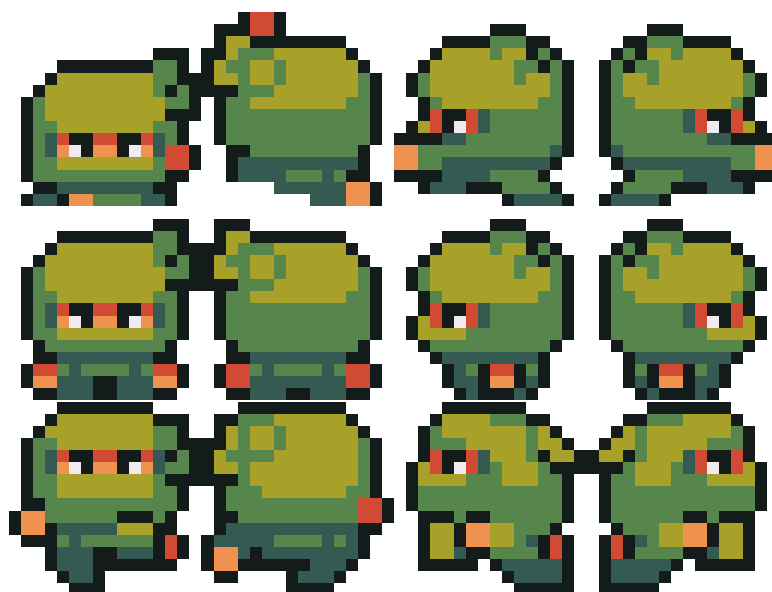


# Desarrollando un videojuego RPG con Unity

Ester Pérez Vázquez

Ciclo Formativo Superior en Desarrollo de Aplicaciones Multiplataforma  
IES Islas Filipinas

---



Junio 2022

Tutora:  
Mar Ruiz

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Descripción del proyecto	2
1.1.1. ¿Qué es Unity?	2
1.1.2. ¿Qué es un videojuego RPG?	3
1.2. Objetivos	3
1.3. Motivación	3
<b>2. Tecnologías y Herramientas utilizadas</b>	<b>4</b>
<b>3. Estimación de Recursos y Planificación</b>	<b>5</b>
3.1. Recursos humanos	5
3.2. Recursos técnicos	6
<b>4. Introducción al desarrollo o Game Design Document</b>	<b>7</b>
4.1. ¿Qué es Game Design Document o GDD?	7
4.2. Concepto del juego	7
4.3. Características	7
4.4. Género	7
4.5. Audiencia objetivo	8
4.6. Jugabilidad	8
4.7. Estilo visual	8
<b>5. Mecánicas de juego</b>	<b>9</b>
5.1. Objetivos	9
5.2. Economía	9
5.3. Controles básicos	9
5.4. Controles específicos	10
5.4.1. Inventario	10
5.4.2. Atacar enemigos	11
5.4.3. Subir las habilidades del personaje	12
5.4.4. Misiones o Quests	13
<b>6. Arquitectura del juego</b>	<b>14</b>
6.1. Diagrama de componentes: Jugador	14
6.2. Diagrama Máquina de estados: Flujo del juego	14
6.3. Diagrama Máquina de estados: Enemigos	15
6.4. Diagrama de Clases: Personaje	15
6.5. Diagrama de Clases: Inventario	16
6.6. Diagrama de Clases: IA	16
6.7. Arquitectura dentro de Unity	17
6.7.1. La ventana de Jerarquía (Hierarchy)	17
6.7.2. La ventana de Inspector	18
6.7.3. Descripción de los componentes de NPC_Herrero	19
<b>7. Requisitos mínimos para el funcionamiento del juego</b>	<b>20</b>
7.1. Windows:	20
7.2. MacOS:	20
7.3. Linux:	20

# Desarrollando un videojuego RPG con Unity

Ester Pérez Vázquez

## 1. Introducción

### 1.1. Descripción del proyecto

#### 1.1.1. ¿Qué es Unity?

Unity es lo que se conoce como un motor de desarrollo o motor de juegos. El término motor de videojuego, game engine, hace referencia a un software el cual tiene una serie de rutinas de programación que permiten el diseño, la creación y el funcionamiento de un entorno interactivo; es decir, de un videojuego. Las funcionalidades típicas que tiene un motor de videojuegos, son las siguientes:

- Motor gráfico para renderizar gráficos 2D y 3D
- Motor físico que simule las leyes de la física
- Animaciones
- Sonidos
- Inteligencia Artificial
- Programación o scripting
- Etc

Unity, antes llamado Unity 3D, es un software que centraliza todo lo necesario para poder desarrollar videojuegos. Es decir, es una herramienta que te permite crear videojuegos para diversas plataformas (PC, videoconsolas, móviles, etc.) mediante un editor visual y programación via scripting, y pudiendo conseguir resultados totalmente profesionales. Prueba de ello son juegos muy famosos creados con éste motor; tales como “Monument Valley”, “Gris” o “Cuphead”. Además, es muy utilizado en la mayoría de desarrollos de videojuegos para móvil. [1]



Figura 1: Faceset del personaje principal, ésta imagen aparece en la Interfaz de Usuario junto con la barra de vida y de maná.

### 1.1.2. ¿Qué es un videojuego RPG?

Los RPG son un género de videojuegos denominados juegos de rol, se caracterizan porque un jugador controla un personaje en un mundo totalmente recreado donde transcurre una historia y hay que ir completando misiones hasta alcanzar el objetivo final. [2].

## 1.2. Objetivos

Los objetivos de este proyecto son aprender todo lo necesario para desarrollar un videojuego de tipo RPG con el motor Unity y con el lenguaje de programación C#.

- Crear al personaje y sus componentes principales como Movimiento, Animaciones, Salud, Maná, Experiencia, Stats y Atributos.
- Crear el mapa utilizando RuleTiles para automatizar la creación de escenarios con Unity y Tilemaps.
- Crear el movimiento de la cámara usando Cinemachine.
- Crear el inventario con el cual se podrá recoger Items, moverlos, usarlos, equiparlos y borrarlos.
- Crear a nuestros NPC y el sistema de Diálogos.
- Crear un sistema de Inteligencia Artificial para los enemigos.
- Crear mecánicas de combate para que nuestro personaje pueda luchar con los enemigos.
- Crear un sistema de Loot Drop para poder recoger Items de enemigos derrotados.
- Crear un sistema de Tienda donde podremos comprar Items.
- Crear un sistema de Crafting con el cual podremos crear Items a base de Recetas.

## 1.3. Motivación

Desde que soy pequeña siempre he estado muy interesada en los videojuegos. He decidido escoger el género RPG porque la saga Golden Sun, lanzada en 2001, fue el primer videojuego de éste género que pude disfrutar con tan solo 8 o 9 años y siempre me llamó la atención lo bien que estaba desarrollado para aquella época, ya que entonces la industria de los videojuegos todavía estaba en pañales. Otra de mis motivaciones a la hora de realizar este proyecto es el impacto social y cultural que tienen los videojuegos en la actualidad, ya que, cada vez más estos son verdaderas obras de arte a nivel guión, animación y audio. Y también, cabe mencionar que está demostrado los beneficios que aportan al desarrollo de la inteligencia. [3]

## 2. Tecnologías y Herramientas utilizadas

- **C#:** es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. C# permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET. C# tiene sus raíces en la familia de lenguajes C.
- **Unity:** es un motor de videojuego multiplataforma creado por Unity Technologies.
- **Visual Studio 2019:** herramienta de entorno de desarrollo integrado (IDE) de Microsoft. La integración de Visual Studio de Unity le permite crear y mantener archivos de proyectos de Visual Studio automáticamente. [4].
- **LaTeX:** es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado de forma especialmente intensa en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas.
- **Git:** Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confianza y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.
- **GitHub:** es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.



Figura 2: Poción de vida, mediante el uso de esta, el jugador recupera puntos de vida

### 3. Estimación de Recursos y Planificación

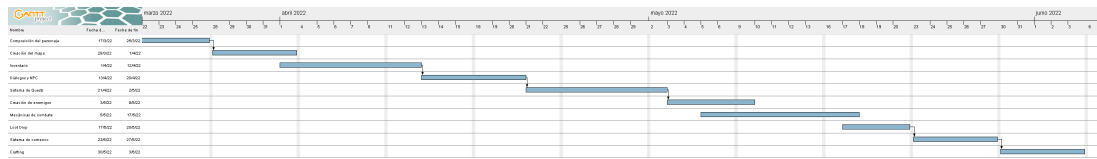


Figura 3: Planificación - Diagrama de Gantt

Los recursos de un proyecto suelen ser los bienes o activos que posee una empresa para poder llevar a cabo el proyecto. El recurso más importante siempre son las personas, pero también dispondrán de equipos tecnológicos y herramientas que aumenten el rendimiento de trabajo.

#### 3.1. Recursos humanos

Normalmente en el desarrollo de un videojuego participa un equipo de personas que desempeñan diferentes roles, los cuales son los siguientes:

- Gestor de Proyectos
- Diseñador Gráfico
- Modelador 2D
- Animador 2D
- Desarrollador
- Diseñador de videojuegos
- Ingeniero de sonido

Como el proyecto actual sólo será llevado a cabo por una persona, se debe calcular el valor de su trabajo económicamente. El salario medio de un Programador de Videojuegos es de 32.100 € brutos por año (alrededor 1.740 € netos por mes) [5]

Esta mensualidad bruta debemos dividirla entre los 22 días laborables de media que tiene un mes y entre las 8 horas de la jornada laboral. El salario bruto a la hora es de 9,89€.

Como nuestro proyecto está estimado a 30 horas, el coste de la mano de obra es 294,9 €.

### 3.2. Recursos técnicos

En este apartado doy a conocer el software y equipo informático necesario para llevar a cabo el proyecto.

EQUIPO INFORMÁTICO	COSTE
Portátil HP Pavilion 13	889,89€
Monitor LG	80,00€
Tableta Huion 1060 Plus	60,00€
Ratón genérico	10,00€
Teclado genérico	15,00€
<b>TOTAL:</b>	1054,89€

Figura 4: Estimación de costes del equipo informático.

SOFTWARE	RAZÓN	COSTE
Unity 2021.2.12f1	Motor de videojuegos	-
Visual Studio 2019	IDE	-
LaTeX	Documentación	-
Adobe PhotoShop	Creación de sprites 2D	24.18€
<b>TOTAL:</b>		24.18€

Figura 5: Estimación de costes de los softwares usados.

## 4. Introducción al desarrollo o Game Design Document

### 4.1. ¿Qué es Game Design Document o GDD?

Se trata de una síntesis de todo lo que va a ser el juego. En él deben constar elementos como: el concepto del juego, la historia, el género, las plataformas, el equipo de producción... Se trata del primer documento que se debe crear a la hora de desarrollar un juego y este siempre se tiene que ir editando según se avanza en el juego. [6]

### 4.2. Concepto del juego

El jugador controla a GreenNinja, quien debe ir completando las misiones para poder completar la primera versión del juego.

### 4.3. Características

El juego se basa en lo siguiente:

- **Sencillez en la historia:** Deberá ser directa y sencilla pero con un final inesperado. La historia se construirá mediante diálogos con los NPC y pequeños detalles en el mapa.
- **Sensación de mundo abierto:** El mapa debe dar la sensación de amplio y diverso, de manera que el jugador tenga la sensación de poder perderse en él.
- **Puzzles divertidos:** Para poder ir avanzando en la historia será necesario ir resolviendo diferentes retos.
- **Historia no lineal:** La historia propondrá una serie de objetivos, éstos no tienen porqué ser alcanzados en un orden determinado, si no que, el jugador puede escoger como hacerlo.

### 4.4. Género

Como ya he mencionado anteriormente, el género principal del que toma elementos es RPG, sin embargo también me gustaría incorporar más adelante los siguientes géneros:

- **RogueLike:** El término nace de "Rogue", uno de los primeros videojuegos que dieron vida a este sistema de juego, aunque no fue el primero en sí. Las características de estos es que sus mapas se van creando de forma aleatoria, es decir, siguiendo un algoritmo que va creando habitaciones de manera que nunca sabemos qué enemigos van a estar presentes o si podemos seguir un destino marcado.[7] La idea sería que en una de las zonas del mapa se le aplicara éste género .
- **Acción-aventura:** Un juego de acción y aventura se puede definir como un juego con una combinación de elementos de un juego de acción y un juego de aventuras, especialmente elementos cruciales como los rompecabezas. Ofrecen una historia, numerosos personajes, un sistema de inventario, diálogo y otras características de los juegos de aventuras.[8]



#### 4.5. Audiencia objetivo

Este juego está dirigido a jugadores de un amplio rango de edades con un tiempo ilimitado o limitado que dedicar al ocio electrónico, ya que gracias a la mezcla de géneros roguelike y RPG, podrán disfrutar del juego tanto de forma esporádica, como de manera inmersiva gracias a la historia.

#### 4.6. Jugabilidad

El juego tendrá una zona del mapa donde el jugador podrá explorar y resolver puzzles necesarios para poder completar la historia, y otra zona donde en cada pantalla se irán generando mazmorras de forma aleatoria. El jugador deberá ir completando cada una de estas zonas derrotando distintos tipos de enemigos. Para ello nos valdremos de los siguientes elementos:

- **Movilidad:** La cámara se situa cerca del personaje pero de forma que podamos ver parte de la escena para poder atacar y esquivar a los enemigos.
- **Puzzles:** Sencillos rompecabezas que forman parte del mapa explorable. Serán necesarios para poder ir avanzando en la historia.
- **Misiones:** Forman parte del mapa explorable, ya que se activarán hablando con los npc. Algunas se podrán completar en éste mapa y otras en la zona de mazmorras.
- **Ataques y hechizos:** Necesarios para poder derrotar a los enemigos.
- **Puntos de habilidad y armas:** El jugador irá recibiendo mejoras en su personaje según vaya subiendo de nivel. También podrá acceder a nuevos tipos de arma según consiga más puntos de habilidad o complete misiones.

#### 4.7. Estilo visual

Basado en el pixel art, de estilo sencillo para encajar con el dinamismo del juego pero con pequeños detalles para incitar la exploración. Los colores serán vivos y las texturas simples.

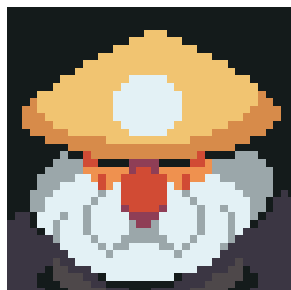


Figura 6: Faceset del NPC mercader, ésta imagen aparece en el panel de conversación.

## 5. Mecánicas de juego

### 5.1. Objetivos

El objetivo del jugador será completar la historia, para ello deberá ir completando los puzzles y las misiones. Con cada misión completada el jugador obtendrá una recompensa que podrán ser: mejoras del personaje o nuevos items. Cada puzzle resuelto permitirá al jugador acceder a un nuevo nivel en la zona de mazmorras. Cada nivel completado desbloqueará distintas animaciones que irán construyendo la historia.

### 5.2. Economía

La economía dentro del juego recae sobre la cantidad de vida, de maná y de experiencia del personaje principal. Es necesario mantener los puntos de vida en un valor positivo o el personaje principal morirá, reiniciándose así su posición en el mapa y teniendo que volver a dirigirse a la zona en la que se encontraba.

Los puntos de maná (o puntos de magia) son necesarios para que el usuario pueda usar las habilidades especiales del personaje principal. Si bien es cierto que no es un requisito totalmente imprescindible para completar el juego, el uso de estas habilidades mejoran la dinámica de juego y ayudan al usuario a superar retos de una forma más cómoda y estratégica.

Al eliminar a cualquier enemigo se obtiene una cantidad variable de puntos de experiencia. Esta experiencia es acumulada hasta que el personaje sube de nivel. Cada vez que el personaje sube de nivel se produce un crecimiento (personalizable por el usuario) en las características pasivas del personaje principal haciéndolo más fuerte, más rápido o más resistente. Siempre será beneficioso para el usuario conseguir la cantidad más alta de experiencia en cada mapa, pues los retos futuros serán más asequibles.

### 5.3. Controles básicos

A continuación se listan los controles predefinidos:

- **Tecla A o flecha izquierda del teclado:** El personaje se desplaza a la izquierda.
- **Tecla D o flecha derecha del teclado:** El personaje se desplaza a la derecha.
- **Tecla S o flecha abajo del teclado:** El personaje se desplaza hacia abajo.
- **Tecla W o flecha arriba del teclado:** El personaje se desplaza hacia arriba.
- **Tecla M:** Dentro del inventario mueve el objeto seleccionado.
- **Tecla E:** Cerca de un NPC inicia la interacción de conversación.
- **Barra de espacio:** En una conversación con un NPC cambia de un bocadillo de texto a otro.
- **Tecla ESC:** Sale del juego.

## 5.4. Controles específicos

### 5.4.1. Inventario

**Abrir el inventario:** El jugador debe “clickar” sobre el botón que contiene la imagen de una mochila en la interfaz del jugador.



Figura 7: Botones para abrir, de izquierda a derecha, el inventario, los stats del personaje y el panel de misiones.

**Equipar un arma:** Primero hay que seleccionar el arma que se quiere equipar, ésta aparecerá como seleccionada mediante un recuadro azul.



Figura 8: Panel de inventario, el arma seleccionada aparece seleccionada con un recuadro de color azul.

Tras esto ya solo queda apretar el botón de equipar y se podrá observar el arma equipada en la interfaz del jugador. Si el arma es de tipo Melee, solo aparecerá la imagen del arma y un recuadro vacío a su lado. Si es de tipo magia, aparecerá el arma y a su lado la imagen del tipo de proyectil.



Figura 9: De izquierda a derecha, arma de tipo Melee y de tipo Magia en la interfaz de usuario.

**Desequipar arma:** Tan sólo habrá que apretar el botón de desequipar.

**Consumir objetos:** El jugador primero debe seleccionar el objeto a consumir, tras esto debe apretar el botón usar.



Figura 10: Panel de inventario con una poción de Maná seleccionada.

#### 5.4.2. Atacar enemigos

**Con un arma de magia:** El jugador primero debe “clickar” sobre un enemigo, debajo de éste aparecerá un círculo blanco. Una vez el enemigo esté seleccionado, el jugador solo tendrá que apretar la barra de espacio para disparar un proyectil.



Figura 11: A la izquierda, imagen de enemigo seleccionado. A la derecha, imagen de jugador disparando un proyectil de magia.

**Con un arma de Melee:** El jugador debe acercarse lo suficiente al enemigo hasta que debajo de este aparezca un círculo rojo de selección, tras esto el jugador ya sólo debe apretar la barra de espacio.

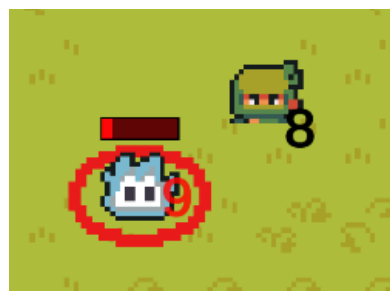


Figura 12: Jugador atacando a un enemigo con un arma Melee.

### 5.4.3. Subir las habilidades del personaje

Cuando nuestro personaje sube de nivel adquiere puntos de habilidad, estos se acumulan en el panel de Stats y permiten incrementar el daño, la defensa, el porcentaje de crítico, el porcentaje de bloqueo y la velocidad. Para ello se deben seguir los siguientes pasos.

**Abrir panel de Stats:** El jugador debe pulsar el botón con el dibujo de un brazo que se encuentra en la interfaz del personaje.

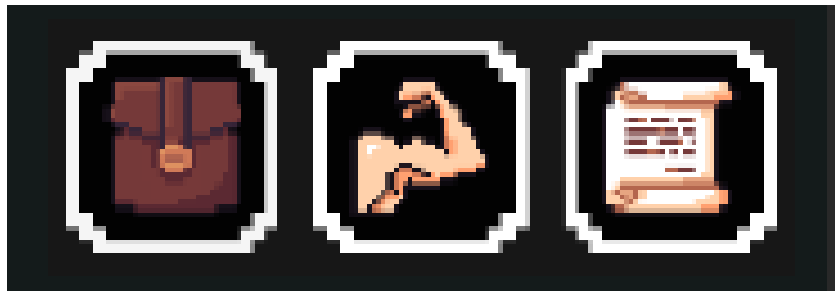


Figura 13: Botones para abrir, de izquierda a derecha, el inventario, los stats del personaje y el panel de misiones.

**Incrementar habilidades:** Con el panel de Stats abierto y tras haber subido de nivel, se podrá observar que en la sección de Atributos tendremos una cantidad de puntos acumulada. Para poder aplicar estos puntos se debe pulsar los botones con el símbolo de “+” que aparecen a la derecha de fuerza, inteligencia y destreza.

Tras pulsar estos botones se podrá observar como las habilidades se han incrementado, exceptuando la de velocidad, ya que en el ejemplo mostrado no hemos pulsado el botón de destreza

Stats			
Daño:	9	Nivel:	2
Defensa:	2	Exp:	2
Crítico:	6%	Exp Req:	4
Bloqueo:	1%		
Velocidad:	5		
Atributos			
Puntos:	3		
Fuerza:	0	+	
Inteligencia:	0	+	
Destreza:	0	+	

Stats			
Daño:	14	Nivel:	2
Defensa:	3	Exp:	2
Crítico:	6,02%	Exp Req:	4
Bloqueo:	1,03%		
Velocidad:	5		
Atributos			
Puntos:	1		
Fuerza:	1	+	
Inteligencia:	1	+	
Destreza:	0	+	

Figura 14: Panel de Stats, a la izquierda, con puntos ya adquiridos pero sin aplicar, a la derecha con los puntos ya aplicados.

#### 5.4.4. Misiones o Quests

Al iniciar el juego, el jugador se encontrará en una habitación con el personaje que se muestra a continuación.

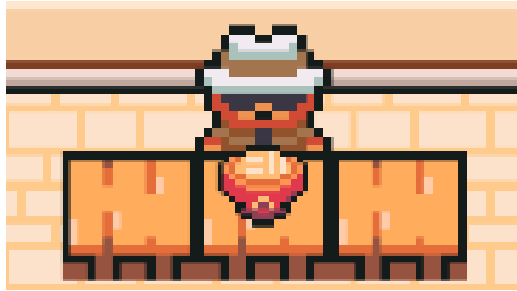


Figura 15: NPC del Sr.M, este personaje nos mostrará las misiones que podemos aceptar.

Al interactuar con él nos mostrará una lista de misiones que podremos aceptar pulsando el botón de “yes”. Se podrá observar que cada misión tiene una descripción del objetivo a cumplir y de sus respectivas recompensas.

Tras aceptar estas misiones, si pulsamos el botón con la imagen de un pergamino que se muestra en la interfaz de usuario (consultar figura 13, pág. 12). Se abrirá el panel de Quests en curso, donde podremos observar el estado de nuestras misiones aceptadas.



Figura 16: A la izquierda, panel de misiones por aceptar que nos muestra el NPC Sr.M. A la derecha el panel de misiones en curso y su estado.

## 6. Arquitectura del juego

### 6.1. Diagrama de componentes: Jugador

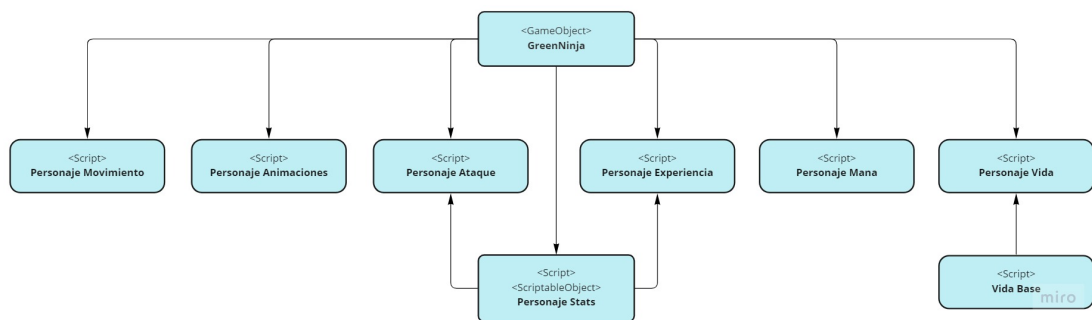


Figura 17: Diagrama del jugador.

### 6.2. Diagrama Máquina de estados: Flujo del juego

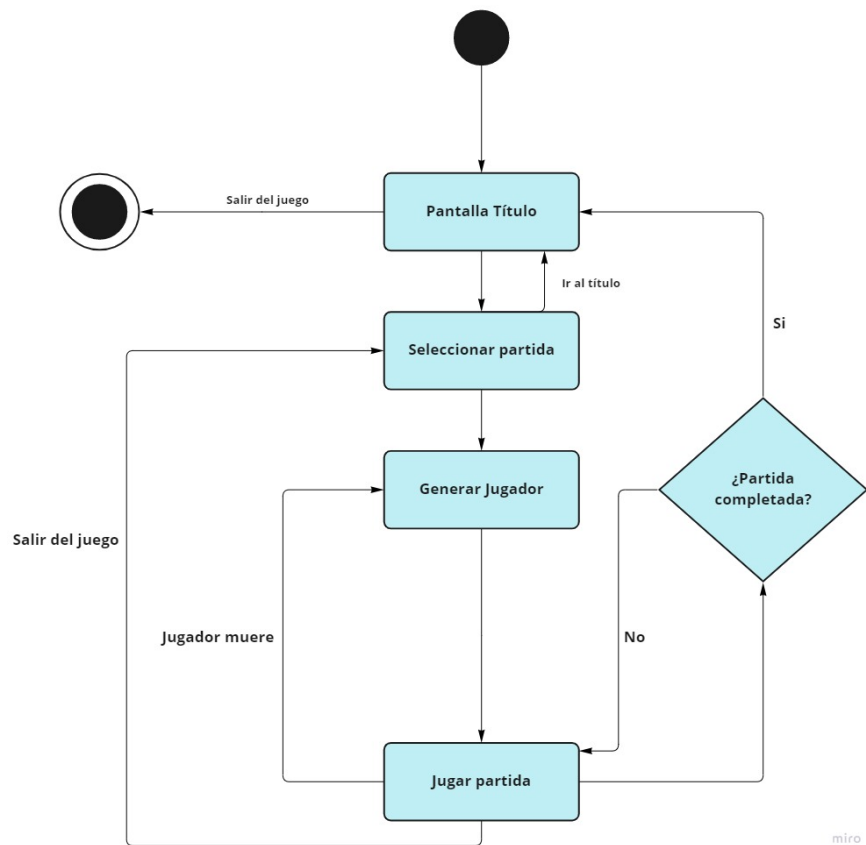


Figura 18: Diagrama Máquina de Estados del Flujo del Juego.

El guardado de la partida se realizaría al salir del juego o al seleccionar la opción de guardar partida en el menú.

### 6.3. Diagrama Máquina de estados: Enemigos

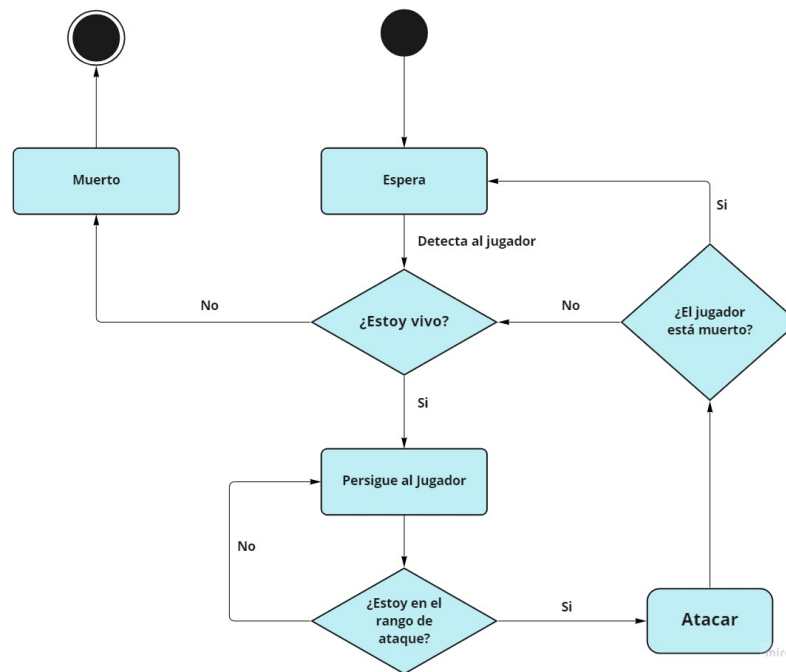


Figura 19: Diagrama Máquina de Estados de los Enemigos.

### 6.4. Diagrama de Clases: Personaje

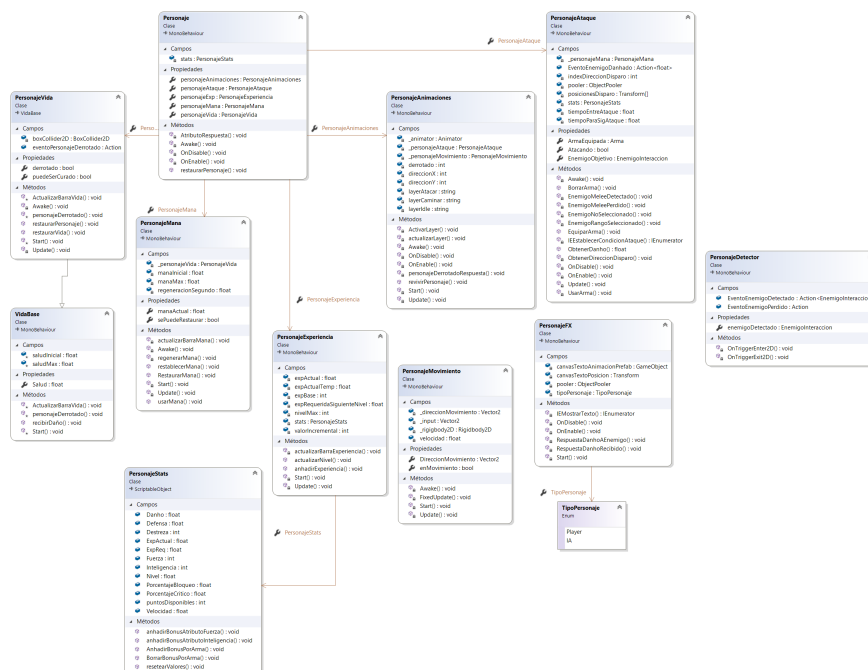


Figura 20: Diagrama de Clases del Personaje.



## 6.5. Diagrama de Clases: Inventario

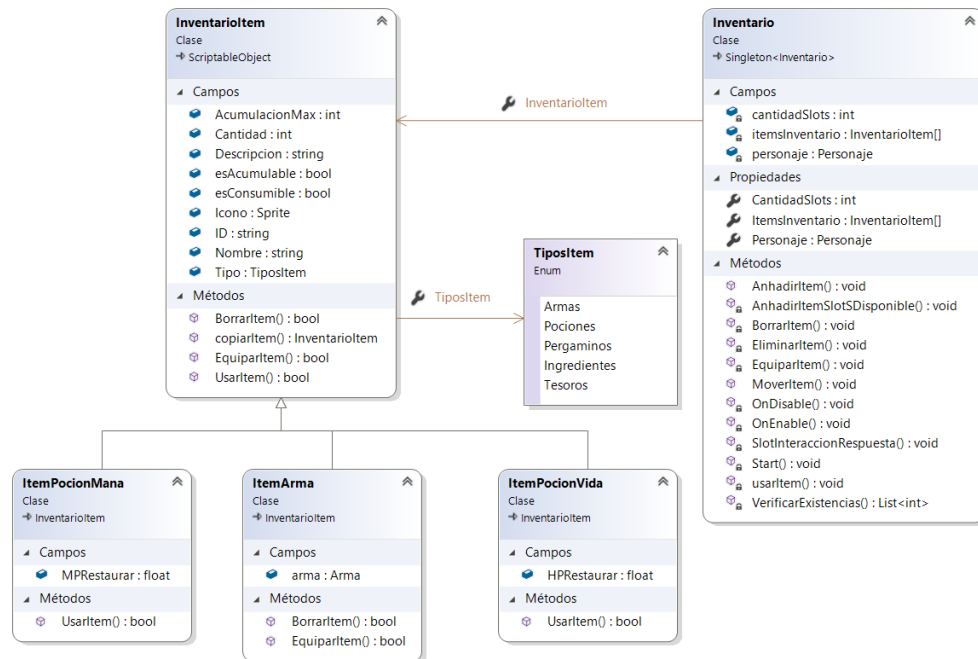


Figura 21: Diagrama de Clases del Inventario.

## 6.6. Diagrama de Clases: IA

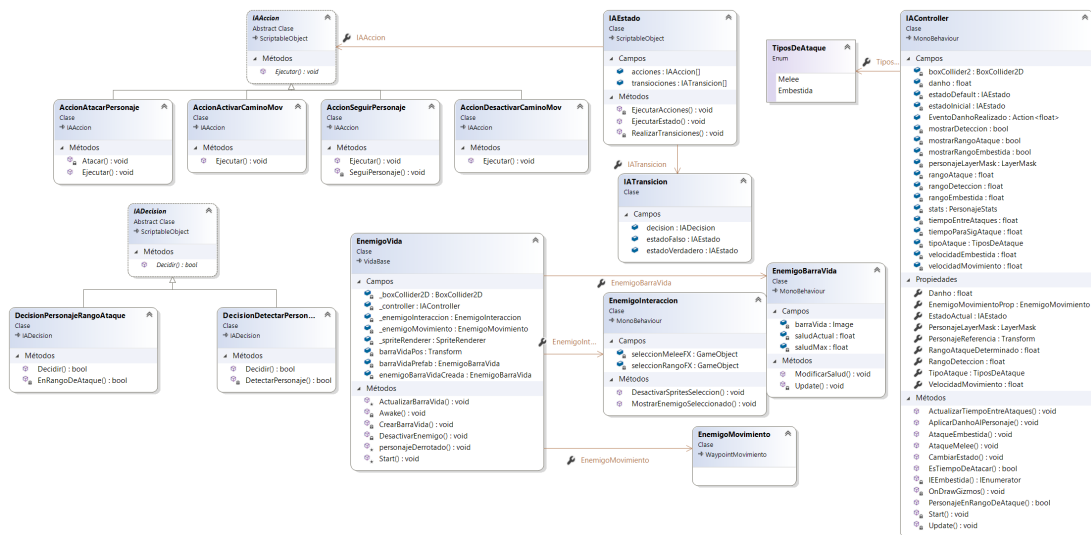


Figura 22: Diagrama de Clases del sistema IA.

## 6.7. Arquitectura dentro de Unity

En esta subsección voy a describir brevemente como funciona Unity mediante uno de los GameObject de mi videojuego.

### ¿Qué es un GameObject?

Los GameObjects son objetos fundamentales en Unity, representan personajes, props, y el escenario. Estos no logran nada por sí mismos pero funcionan como contenedores para Components, que implementan la verdadera funcionalidad. Por ejemplo, un objeto Light es creado al adjuntar un componente Light a un GameObject. [9]

#### 6.7.1. La ventana de Jerarquía (Hierarchy)

La Hierarchy contiene cada GameObject de la escena actual. Algunos de estos son instancias directas de archivos de asset, como modelos 3D, y otras son instancias de Prefabs, objetos personalizados que van hacer gran parte del juego. Se puede seleccionar objetos en la jerarquía y luego arrastrar un objeto a otro para hacer uso de Parenting (Un gameObject padre puede tener varios hijos de distintos tipos). Como hay objetos que están siendo agregados y eliminados en la escena, estos van a aparecer y desaparecer de la jerarquía también. Por defecto los GameObjects se mostrarán en la ventana Hierarchy en el orden en que se crean. Es posible reordenar los GameObjects arrastrándolos arriba o abajo, o haciéndolos hijos o padres.[10]

En el caso de este proyecto, la jerarquía es la mostrada en la siguiente figura. En esta podemos observar como del GameObject NPC\_Herrero, el cual se muestra seleccionado en azul, cuelgan a su vez un componente Canvas, que a su vez éste, tiene como hijo una Image y ésta un Text.

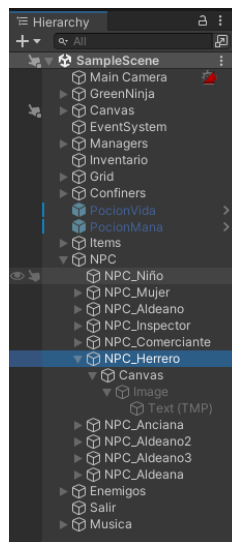


Figura 23: Jerarquía del videojuego donde se muestra desplegado el gameObject NPC\_Herrero.

### 6.7.2. La ventana de Inspector

El Inspector es usado para ver y editar propiedades de objeto y también preferencias y otros ajustes dentro de Unity.

Cuando seleccionamos un GameObject en la Jerarquía o Scene View, el Inspector va a mostrar las Properties de todos los Components y Materials en el objeto y nos permitirá editarlos. Para un Script component personalizado, las variables públicas de ese script son mostradas en el Inspector y pueden ser vistas y editadas como las propiedades de los components integrados de Unity. Esto le permite establecer parámetros y valores por defecto en su scripts fácilmente sin modificar el código. [11]

Si observamos el inspector de nuestro NPC\_Herrero vamos a notar que tiene 5 componentes, un Transform, un Sprite Renderer, dos Colliders y un Script. Si seleccionáramos uno de los GameObjects hijo podríamos ver que los componentes del Inspector cambian.

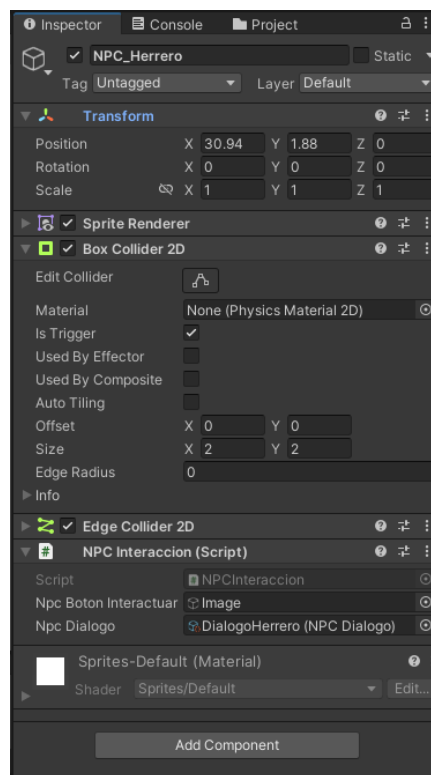


Figura 24: Ventana Inspector de NPC\_Herrero.

### 6.7.3. Descripción de los componentes de NPC\_Herrero

- **Transform:** Es imposible crear un GameObject en Unity sin un Transform Component. El Transform Component es uno de los más importantes Components, ya que todas las propiedades Transform del GameObject están habilitadas por sus usos en este Component. Define la posición, rotación, y escala del GameObject en el mundo/Scene View del juego. Si un GameObject no tuviera un Transform Component, sería nada más que alguna información en la memoria del ordenador.[12]
- **Sprite Renderer:** Este componente permite mostrar imágenes como Sprites para su uso en escenas 2D y 3D. En este también podemos decidir en que layer se mostrará la imagen para así poder dar o quitar profundidad sobre la escena y otros Sprites.
- **Colliders 2D:** Definen una forma aproximada de un objeto que es utilizado por el motor de física para determinar colisiones con otros objetos. También se pueden usar como “triggers” de manera que cuando nuestro personaje se acerca al collider sucede una acción o evento.
- **Script:** Es el código que mueve la lógica de cada GameObject. En el Script NPCInteraccion de nuestro GameObject NPC\_Herrero se puede observar como aparecen dos campos, Npc Boton interactuar y Npc Dialogo. Estos dos campos hacen referencia a dos variables dentro de nuestra clase NPCInteraccion. La variable Npc Boton Interactuar es de tipo GameObject y vemos que hace referencia a la Image hija dentro de NPC\_Herrero. Por otro lado, la variable Npc Dialogo es un objeto de tipo NPCDialogo, que a su vez hereda de ScriptableObject, lo que permite generar distintos objetos de esta clase mediante la interfaz de Unity.

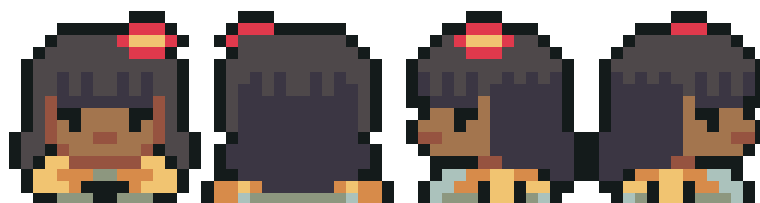


Figura 25: Sprites del NPC Lola.

## 7. Requisitos mínimos para el funcionamiento del juego

### 7.1. Windows:

-	MÍNIMO	RECOMENDADO
Sistema Operativo	Windows 7	Windows 10
Procesador	Intel Core 2	Intel Core i5
Memoria	4 GB de RAM	8GB de RAM
Gráficos	GeForce 9800GTX+ (1GB)	GeForce GTX 560
Almacenamiento	9GB de espacio disponible	9GB des espacio disponible

Figura 26: Tabla de requisitos del sistema: Windows.

### 7.2. MacOS:

-	MÍNIMO	RECOMENDADO
Sistema Operativo	Mac OS 10.13 (64bit)	Mac OS 10.13 (64bit)
Procesador	Intel Core i3	Intel Core i5
Memoria	4 GB de RAM	8GB de RAM
Gráficos	GeForce GTX 470	GeForce GTX 570
Almacenamiento	9GB de espacio disponible	9GB de espacio disponible

Figura 27: Tabla de requisitos del sistema: MacOS.

### 7.3. Linux:

-	MÍNIMO	RECOMENDADO
Sistema Operativo	Ubuntu 16.04 LTS (64bit)	Ubuntu 16.04 LTS (64bit)
Procesador	Intel Core 2 Duo E5200	Intel Core i5
Memoria	4 GB de RAM	8GB de RAM
Gráficos	GeForce 9800GTX+ (1GB)	GeForce GTX 560
Almacenamiento	9GB de espacio disponible	9GB de espacio disponible

Figura 28: Tabla de requisitos del sistema: Linux.

## Referencias

- [1] MasterD, “¿qué es unity y para qué sirve?” April 2022. [Online]. Available: <https://www.masterd.es/blog/que-es-unity-3d-tutorial>
- [2] Geekno, “Rpg: ¿qué significa rpg?: Características de los juegos rpg,” 2022. [Online]. Available: <https://www.geekno.com/glosario/rpg>
- [3] ITSoftware, “Importancia de los videojuegos para formar personalidad e inteligencia,” 2020. [Online]. Available: <https://itsoftware.com.co/content/importancia-de-los-videojuegos/>
- [4] MicrosoftDocs, “Un paseo por c#: información general,” 2022. [Online]. Available: <https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>
- [5] jobted, “¿cuánto cobra un programador de videojuegos?” [Online]. Available: <https://www.jobted.es/salario/programador-videojuegos#:~:text=El%20salario%20medio%20de%20un,salario%20medio%20anual%20en%20Espa%C3%B1a.>
- [6] geekno, “¿qué significa gdd?” Feb 2020. [Online]. Available: <https://www.geekno.com/glosario/gdd>
- [7] juegosadn, “¿qué es un roguelike?” Sept 2020. [Online]. Available: <https://juegosadn.es/que-es-un-rogue-lite-y-un-rogue-like-ar-3790/#que-son-los-roguelike>
- [8] hmong, “Juego de acción y aventura.” [Online]. Available: <https://hmong.es/wiki/Action-adventure>
- [9] unity, “GameObject.” [Online]. Available: <https://docs.unity3d.com/es/530/Manual/class-GameObject.html#:~:text=Los%20GameObjects%20son%20objetos%20fundamentales,componente%20Light%20a%20un%20GameObject.>
- [10] —, “La ventana de jerarquía.” [Online]. Available: <https://docs.unity3d.com/es/530/Manual/Hierarchy.html>
- [11] —, “La ventana de inspector.” [Online]. Available: <https://docs.unity3d.com/es/530/Manual/UsingTheInspector.html>
- [12] —, “Introducción a los componentes.” [Online]. Available: <https://docs.unity3d.com/es/530/Manual/Components.html>