



Instituto Politécnico Nacional
Escuela Superior de Cómputo



Ingeniería en Inteligencia Artificial, Análisis y diseño de sistemas
Sem: 2024-2, 4BM2, TMP(Nombre de la actividad), Fecha: 27-06-24

ECOS DEL ALMA: PROTOTIPO DE JUEGO ROGUE LIKE CON INTERACCIÓN JUGADOR-NPC DINÁMICA

Presentan

Angeles López Erick Jesse¹
Corona Anchondo José Antonio²
Espinosa Martínez José Carlos³
Esquivel García Thania Paola⁴

Resumen:

Palabras Clave:

¹eangeles11700@alumno.ipn.mx

²jcoronaa1900@alumno.ipn.mx

³jespinosam1900@alumno.ipn.mx

⁴tesquivelg2000@alumno.ipn.mx

Contents

1	Introducción	3
2	Estado del arte	4
3	Justificación	5
4	Objetivo general	6
5	Objetivos particulares	6
6	TMP(Capitulos de contexto)	7
7	Análisis	8
7.1	Metodología	8
7.2	Tecnología	8
7.2.1	Motor de desarrollo Unity	8
7.2.2	Lenguaje de programación C#	9
7.2.3	Lenguaje de programación Prolog	9
7.2.4	MongoDB	9
7.2.5	Star UML	10
7.2.6	Pixilart	10
7.3	Requerimientos	11
7.3.1	Requerimientos funcionales	11
7.3.2	Requerimientos no funcionales	12
7.3.3	Requerimientos mínimos de sistema	14
8	Diseño	15
8.1	Diagrama de contexto	15
8.2	Diagrama de flujo de datos	15
8.3	Diagrama de casos de uso	16
8.3.1	Especificaciones de casos de uso del jugador(personaje)	17
8.4	Diagrama de clases	27
8.5	Diagrama de secuencia	28
8.6	Diagrama de estados	28
8.7	Diagrama de actividades	28
9	Prototipo	29
10	Referencias Bibliográficas	30

1 Introducción

2 Estado del arte

Algunos juegos y aplicaciones similares son:

1. Videojuego *Hades* desarrollado por Supergiant Games [1].
2. Videojuego *Celeste* desarrollado por Maddy Makes Games [2]
3. Videojuego *Hollow Knight* desarrollado por Team Cherry [3]
4. Videojuego *Coffee Talk* desarrollado por Toge Productions [4]
5. Aplicación *character.ai* desarrollada por Character Technologies, Inc. [5]

Producto	Características
Hades	Juego <i>rogue like</i> con niveles generados proceduralmente conectando diferentes habitaciones.
Celeste	Juego de plataformas con jugabilidad invariable, conversación ocasional con NPC y mecanicas de movimiento avanzadas.
Hollow knight	Juego <i>metroidvania</i> con poca interacción conversacional, mecanicas de movimiento y combate avanzada.
Coffee talk	Juego de genero novela visual y aventura conversacional, las respuestas (limitadas por el juego) definen las relaciones con los NPC.
character.ai	Aplicación online para diseñar o conversar con bots inteligentes y personificados unicamente mediante lineas de texto.

Table 1: Resumen de juegos y aplicaciones similares

3 Justificación

4 Objetivo general

Desarrollar el prototipo de un videojuego *rogue like* con generación de niveles procedurales e interacción jugador-NPC desde el teclado mediante análisis de texto, simulación de emociones y toma de decisiones.

5 Objetivos particulares

- Desarrollar un generador de niveles procedurales con habitaciones pre-diseñadas estilo *dungeon*.
- Diseñar, para los NPC, factores de personalidad y flujos conversacionales.
- Desarrollar un modelo de procesamiento de lenguaje natural para analizar respuestas del jugador.
- Recopilar conversaciones y respuestas entre el jugador y los NPC.

6 TMP(Capitulos de contexto)

7 Análisis

7.1 Metodología

La metodología para el proyecto es el modelo de entrega por etapas. Este modelo nos permite planificar y diseñar componentes de forma individual siempre orientados a los mismos objetivos.

El videojuego puede tener productos funcionales en las primeras etapas de desarrollo. Es por esto por lo que el modelo de entrega por etapas maximiza la productividad, permitiendo desarrollar elementos adicionales en cada una de sus etapas.

Estas implementaciones pueden ser desde niveles adicionales, mecánicas nuevas, correcciones y expansiones del mismo proyecto, permitiendo diversidad y adaptabilidad a pequeños cambios.

7.2 Tecnología

7.2.1 Motor de desarrollo Unity

Unity es un motor y una plataforma de desarrollo de videojuegos para diseñar contenido audiovisual en 2D y 3D. Permite a programadores, artistas, arquitectos, diseñadores, cineastas, etc. diseñar, construir y desplegar sus propios proyectos en diferentes dispositivos [6].

Se opta por Unity sobre otras herramientas populares en el mercado como lo son Godot, Unreal Engine y Game Maker por las siguientes razones:

- Unity es el motor estandarizado para el desarrollo de proyectos nuevos, con una amplia comunidad y numerosos recursos disponibles.
- El juego requiere de *sprites* estilo *pixel art* que pueden ser segmentados y animados fácilmente mediante el IDE de Unity.
- El juego no requiere mucho procesamiento en los gráficos, por lo que Unity resulta una opción más cómoda y eficiente frente a Unreal Engine, que está más orientado a gráficos de alta calidad y realismo.
- Unity ofrece una excelente documentación y soporte técnico, lo que facilita la resolución de problemas y la implementación de características avanzadas.
- El editor visual de Unity es intuitivo y facilita la creación de niveles y la implementación de mecánicas de juego complejas sin requerir conocimientos avanzados en programación.
- Unity proporciona un excelente sistema de física y colisiones que puede ser aprovechado para crear interacciones detalladas y precisas en el juego.

7.2.2 Lenguaje de programación C#

Lenguaje de programación utilizado comúnmente en Unity para el desarrollo de videojuegos. Es un lenguaje orientado a objetos con un sistema de gestión automática de memoria [7].

7.2.3 Lenguaje de programación Prolog

Lenguaje de programación basado en declaración de hechos, establecimiento de reglas e inferencia de preguntas. Permite generar una base de conocimientos relacionando diferentes hechos mediante secuencias lógicas enlazadas [8].

Su uso sera como una librería a la cual podremos acceder desde C# para generar la base de conocimientos y la generación de inferencias.

7.2.4 MongoDB

MongoDB es un sistema gestor de base de datos NoSQL orientado a documentos con estructuras BSON y esquematización dinámica.

- Incluye medidas de seguridad robustas:
 - **Autenticación:** Registro mediante credenciales de Atlas, GitHub o Google, lo que garantiza un acceso controlado y seguro a la base de datos.
 - **Autorización:** Creación y asignación de roles y permisos para controlar el acceso y las operaciones que cada usuario puede realizar.
 - **Auditoría:** Monitoreo detallado de acciones en el entorno de MongoDB, permitiendo rastrear cambios y accesos para garantizar la seguridad y el cumplimiento de normas.
 - **Encriptación:** Protección de datos tanto en reposo como en tránsito, asegurando que la información sensible esté siempre protegida contra accesos no autorizados.
- Ofrece tiempos de consulta reducidos en comparación con sistemas relacionales, lo que mejora significativamente el rendimiento en aplicaciones con grandes volúmenes de datos.
- Escalabilidad horizontal y vertical, permitiendo ajustar la capacidad de la base de datos según las necesidades del proyecto. Esto es especialmente útil para aplicaciones que requieren manejar grandes cantidades de datos y usuarios.
- Herramientas de transcripción de modelos SQL a NoSQL, facilitando la migración y la integración de bases de datos existentes.

- Conectividad con diferentes herramientas y sistemas de gestión, proporcionando flexibilidad y facilitando la integración con otros componentes del ecosistema tecnológico del proyecto.
- Integración con frameworks de desarrollo populares, lo que facilita su uso en diversas aplicaciones y entornos de desarrollo [9].

7.2.5 Star UML

Herramienta de modelado UML desarrollado por MKLab. Es una de las herramientas UML más populares en el mundo. Se ha descargado más de 4 millones de personas y se utiliza en más de 150 países.

Es compatible con los principales lenguajes de programación como Java, C# y C ++. Puede generar códigos fuente de sus modelos o construir un modelo a partir de código fuente mediante ingeniería inversa [10].

7.2.6 Pixilart

Aplicación web y red social que permite diseñar, modificar, publicar y compartir imágenes estilo *pixel art* de forma gratuita [11].

Esta plataforma nos permite crear grupos privados donde se puedan compartir y editar diferentes diseños. Cuenta con un amplio abanico de herramientas para colorear y dibujar en cuadrícula, permitiendo formatos de alta calidad.

7.3 Requerimientos

7.3.1 Requerimientos funcionales

Respecto a la jugabilidad

- RF1 Al iniciar, el juego muestra el menú principal con opciones para cargar partida, borrar partida, salir y acceder a la configuración.
- RF2 El jugador controla las acciones y las interacciones del personaje jugable.
- RF3 Al inicio, el juego sitúa al jugador y a un número exacto de NPC en el *respawn*.
- RF4 El jugador comienza un intento de progresión.
- RF5 Cuando el jugador inicia el intento o termina un nivel previo, el juego genera de forma procedural el siguiente nivel, compuesto por un número finito de habitaciones con obstáculos.
- RF6 Cuando el jugador termina el último nivel, regresa al jugador al *respawn*.
- RF7 Al finalizar cada nivel, el juego presenta una habitación con un NPC.
- RF8 Cuando el personaje agota sus vidas, reaparece con el último NPC con el que interactuó y entra en modo “after” (conservando los mismos controles), y sus estadísticas se determinan con base en el flujo de conversación.
- RF9 Si el personaje encuentra su lugar de muerte en modo “after”, regresa al modo normal.
- RF10 Si el personaje muere en modo “after”, vuelve al *respawn* con el último NPC con el que interactuó ahí, y sus estadísticas se ajustan de acuerdo con el flujo de conversación.

Respecto a la configuración

- RF11 El jugador puede configurar manualmente todos los controles o elegir la configuración predeterminada.
- RF12 Se permite al jugador controlar el volumen general del juego, los efectos y la música.
- RF13 Al terminar la interacción con un personaje, el juego guarda los datos de la partida.
- RF14 Al volver al menú principal, el jugador guarda los datos de la partida.

Respecto a los controles

- RF15 **Movimiento lateral:** El personaje puede desplazarse a lo largo del plano 2D.
- RF16 **Salto:** El personaje efectúa un salto vertical independiente del movimiento lateral, limitado por colisiones.
- RF17 **Dash:** El personaje realiza un impulso rápido en la dirección elegida por el jugador, que tiene un período de enfriamiento antes de poder reutilizarse.

Respecto a la interacción con el entorno

- RF18 **Conversación:** El personaje puede iniciar conversaciones con NPC.
- RF19 **Finalizar conversación:** El jugador puede cerrar la ventana de diálogo con el NPC.
- RF20 **Cambio de escenario:** El personaje puede interactuar con puertas (o agujeros) para entrar, salir o cambiar de escenario.
- RF21 **Sistema de vidas:** El personaje pierde salud al colisionar con obstáculos y la recupera al recoger objetos de curación.

Respecto a la interacción con el sistema inteligente

- RF22 Cuando el jugador interactúa con un NPC, este responde según el flujo de conversación actualizado.
- RF23 El jugador introduce su respuesta en la conversación mediante el teclado.
- RF24 Con la respuesta del jugador, el juego procesa el mensaje, genera una respuesta y actualiza el flujo de conversación.
- RF25 El jugador tiene un número máximo de respuestas por encuentro con cada NPC.

7.3.2 Requerimientos no funcionales

- RNF1 **Seguridad de Acceso:** Solo los jugadores registrados podrán acceder al juego, asegurando una experiencia segura y personalizada.
- RNF2 **Personalización Basada en el Perfil del Jugador:** El contenido y las opciones disponibles en el juego variarán según el progreso y las acciones del jugador.

- RNF3 **Conexión a Internet para Funcionalidades del sistema inteligente:** Se requerirá conexión a internet para acceder al servidor que contiene el sistema inteligente, permitiendo la interacción dinámica con NPC. El juego no incluirá características multiplayer online.
- RNF4 **Mensajes de Error Claros:** Los mensajes de error serán informativos, ayudando a los jugadores a entender y resolver problemas rápidamente.
- RNF5 **Diseño Adaptable a Diferentes Resoluciones y Plataformas de Escritorio:** El juego, diseñado exclusivamente para escritorio (Windows y macOS), adaptará su interfaz y gráficos a diferentes resoluciones de pantalla.
- RNF6 **Usabilidad:** El juego será intuitivo y fácil de navegar, ofreciendo una experiencia agradable para un amplio rango de jugadores.
- RNF7 **Terminología Amigable:** Se utilizará un lenguaje claro y accesible, evitando jerga complicada para facilitar la comprensión del juego.
- RNF8 **Tiempo de Generación de Contenido Dinámico:** Contenidos como diálogos o eventos se generarán en menos de 10 segundos, manteniendo el flujo del juego.
- RNF9 **Generación Simultánea de Contenidos:** El juego será capaz de generar múltiples elementos de juego de forma simultánea sin degradar el rendimiento.
- RNF10 **Integridad del Contenido Generado:** Se asegurará que el contenido generado, como diálogos o eventos, sea relevante y libre de errores significativos.
- RNF11 **Feedback Visual Durante Cargas:** Se proporcionarán indicadores de progreso durante las cargas para mantener informados a los jugadores.
- RNF12 **Accesibilidad y Configuración de Teclado:** El juego permitirá a los usuarios configurar las teclas de acceso rápido y ajustar la sensibilidad del teclado.
- RNF13 **Latencia Baja en la Entrada del Teclado:** Se garantizará una latencia mínima en la respuesta a las entradas del teclado.
- RNF14 **Soporte para Atajos de Teclado:** El juego incluirá atajos de teclado intuitivos para acciones comunes.
- RNF15 **Estilo Visual Amigable y Pixel Art:** El diseño del juego, basado en pixel art, promoverá un estilo visual amigable y atractivo, adecuado para una amplia audiencia.

RNF16 **Identificación Clara de Objetos y Elementos:** Tanto en el apartado visual como sonoro, los objetos y elementos importantes del juego serán fácilmente identificables, mejorando la experiencia de juego.

RNF17 **Juiciness en la Jugabilidad:** El juego incorpora elementos de "juiciness", ofreciendo retroalimentación inmediata y satisfactoria a las acciones del jugador a través de efectos visuales, sonoros y mecánicos que hacen que la experiencia de juego sea más dinámica y gratificante.

7.3.3 Requerimientos mínimos de sistema

Windows	
Sistema operativo	Windows 7 o mas reciente.
Procesador	Intel Core i3 M380.
Memoria	2GM de RAM.
Gráficos	Intel HD 4000.
DirectX	Versión 10.
Almacenamiento	1.5 GB de espacio disponible.

Table 2: Requerimientos mínimos de sistema para Windows.

Macos	
Sistema operativo	Lion 10.7.5, 32/64 bits o más reciente.
Procesador	Intel Core i3 M380.
Memoria	2GM de RAM.
Gráficos	OpenGL 3.0 o superior.
Almacenamiento	1.5 GB de espacio disponible.

Table 3: Requerimientos mínimos de sistema para Macos.

8 Diseño

8.1 Diagrama de contexto

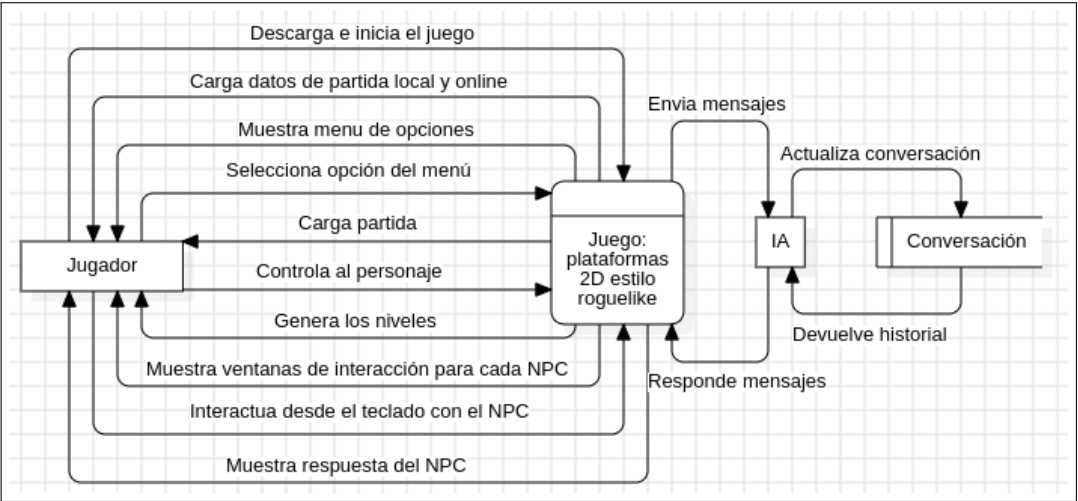


Figure 1: Diagrama de contexto

8.2 Diagrama de flujo de datos

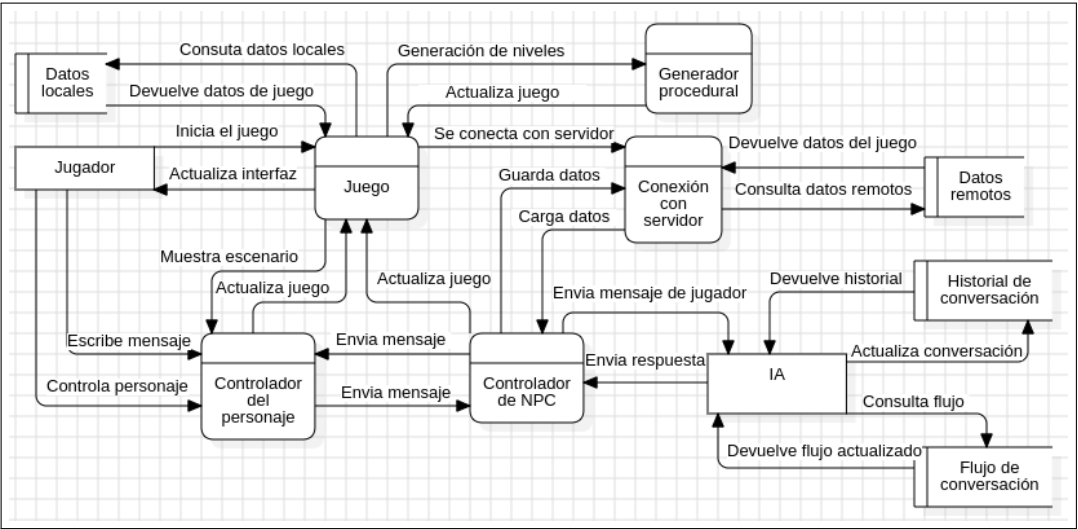


Figure 2: Diagrama de flujo de datos

8.3 Diagrama de casos de uso

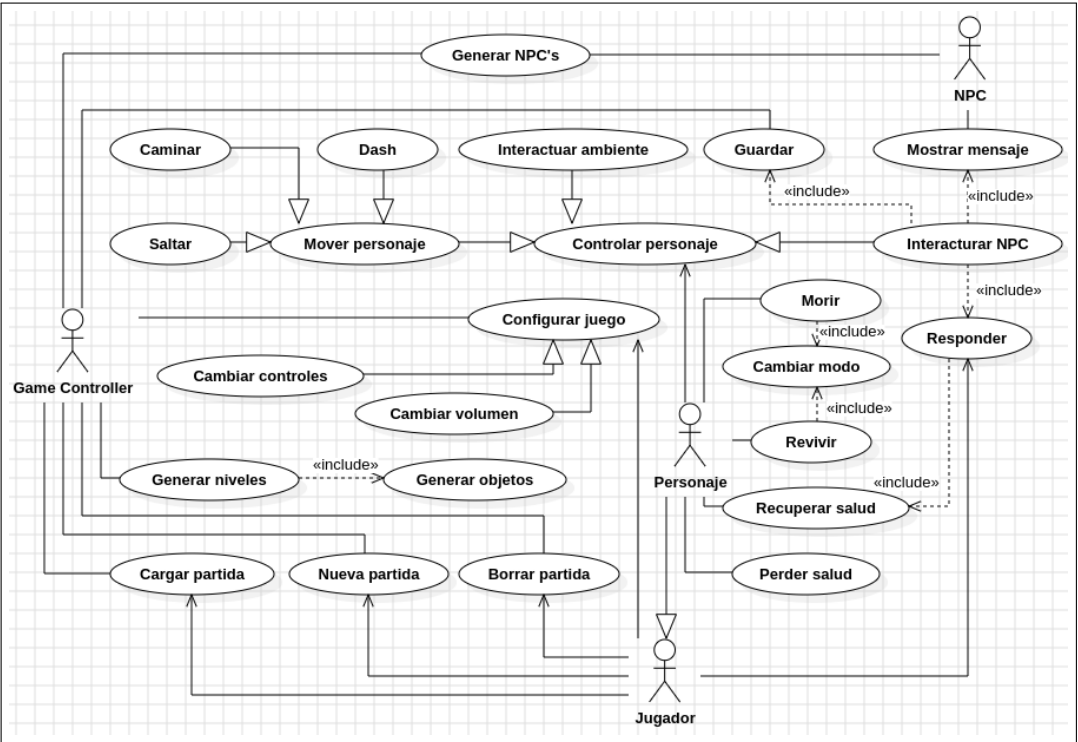


Figure 3: Diagrama de casos de uso

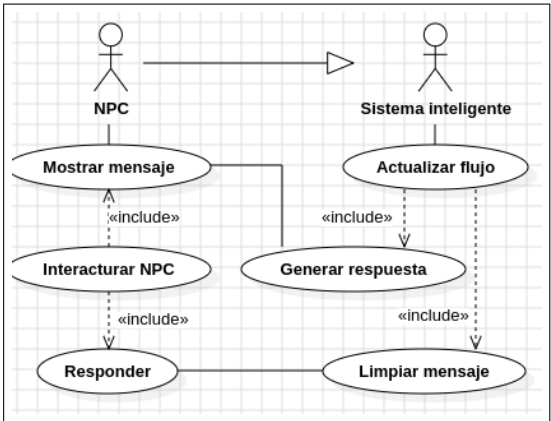


Figure 4: Diagrama de casos de uso

8.3.1 Especificaciones de casos de uso del jugador(personaje)



Figure 5: Caso de uso: Nueva partida

CASO DE USO	Nueva partida	
ACTOR	Jugador	
DESCRIPCIÓN	El jugador inicia una nueva partida en un ‘slot’ disponible.	
PRECONDICIÓN	Ninguna	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Presiona el botón de ‘Nueva partida’ en el menú principal. 3. Selecciona un ‘slot’ disponible.	2. Muestra ‘slots’ de almacenamiento. 4. Crea datos de partida e inicia el juego.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Sin ‘slots’ de almacenamiento: Cancelar.	1. Presiona el botón de ‘Nueva partida’ en el menú principal. 3. Selecciona un ‘slot’ no disponible. 5. Selecciona la opción de cancelar.	2. Muestra ‘slots’ de almacenamiento. 4. Mensaje de advertencia para sobrescribir datos o cancelar. 6. Regresa al menú principal.
FLUJO ALTERNO	ACTOR	SISTEMA
2. Sin ‘slots’ de almacenamiento: Cancelar.	1. Presiona el botón de ‘Nueva partida’ en el menú principal. 3. Selecciona un ‘slot’ no disponible. 5. Selecciona la opción de sobrescribir.	2. Muestra ‘slots’ de almacenamiento. 4. Mensaje de advertencia para sobrescribir datos o cancelar. 6. Borra datos de partida pasada, crea datos de partida e inicia el juego.
POSTCONDICIÓN	Ninguna	

Table 4: Descripción del caso de uso: Nueva partida.



Figure 6: Caso de uso: Cargar partida

CASO DE USO	Cargar partida	
ACTOR	Jugador	
DESCRIPCIÓN	El jugador carga los datos de la partida guardada e inicia el juego.	
PRECONDICIÓN	Nueva partida	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Selecciona el ‘slot’ de partida.	2. Carga datos de partida, e inicia el juego.
POSTCONDICIÓN	Ninguna	

Table 5: Descripción del caso de uso: Cargar partida



Figure 7: Caso de uso: Borrar partida.

CASO DE USO	Borrar partida	
ACTOR	Jugador	
DESCRIPCIÓN	El jugador elimina los datos guardados de una partida.	
PRECONDICIÓN	Nueva partida	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Selecciona el botón de eliminar ‘slot’ de partida. 3. Selecciona eliminar los datos.	2. Mensaje de advertencia para eliminar o cancelar. 4. Elimina permanente los datos de partida del ‘slot’.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Cancelar.	1. Selecciona el botón de eliminar ‘slot’ de partida. 3. Selecciona cancelar.	2. Mensaje de advertencia para eliminar o cancelar. 4. Regresa al menú principal.
POSTCONDICIÓN	Ninguna	

Table 6: Descripción del caso de uso: Eliminar partida

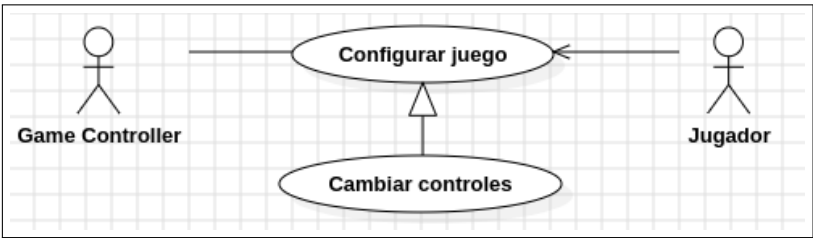


Figure 8: Caso de uso: Cambiar controles

CASO DE USO	Cambiar controles	
ACTOR	Jugador	
DESCRIPCIÓN	El jugador configura los controles para las acciones del personaje.	
PRECONDICIÓN	Ninguna	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Selecciona el menú de configuración. 3. Selecciona el sub menú de controles. 5. Selecciona una acción. 7. Presiona un botón en el tiempo dado.	2. Muestra configuración de audio y de controles. 4. Muestra un listado de acciones y su botón/tecla asignada. 6. Inicia un contador esperando respuesta del jugador. 8. Configura la acción al botón seleccionado.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Omisión.	1. Selecciona el menú de configuración. 3. Selecciona el sub menú de controles. 5. Selecciona una acción. 7. No presiona un botón en el tiempo dado.	2. Muestra configuración de audio y de controles. 4. Muestra un listado de acciones y su botón/tecla asignada. 6. Inicia un contador esperando respuesta del jugador. 8. Mantiene la configuración actual de la acción.
POSTCONDICIÓN	Ninguna	

Table 7: Descripción del caso de uso: Cambiar controles

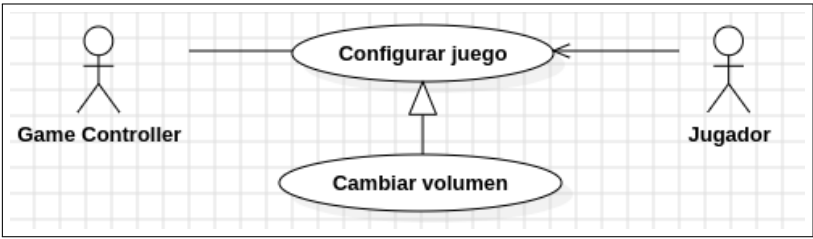


Figure 9: Caso de uso: Cambiar volumen

CASO DE USO	Cambiar volumen	
ACTOR	Jugador	
DESCRIPCIÓN	Se configuran los volúmenes del juego	
PRECONDICIÓN	Ninguna	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Selecciona el menú de configuración. 3. Selecciona el sub menú de volumen. 5. Selecciona y desplaza la barra de volumen específica.	2. Muestra configuración de audio y de controles. 4. Muestra barras deslizables de volumen. 6. Actualiza el volumen seleccionado.
POSTCONDICIÓN	Ninguna	

Table 8: Descripción del caso de uso: Cambiar volumen

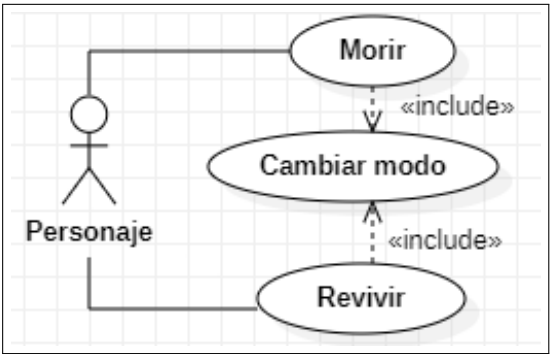


Figure 10: Caso de uso: Cambiar modo

CASO DE USO	Cambiar modo	
ACTOR	Personaje (Jugador)	
DESCRIPCIÓN	El personaje tiene un segundo intento después de morir y lo recupera cuando revive	
PRECONDICIÓN	Morir o Revivir	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. El personaje muere. 2. El personaje encuentra el lugar donde falleció.	2. El personaje reaparece con el ultimo NPC con el que interactuó (Sus estadísticas están determinadas por el flujo conversacional del NPC) en modo 'Afterlife'. 3. El personaje recupera sus estadísticas iniciales y continua el juego.
FLUJO ALTERNO	ACTOR	SISTEMA
1. El jugador fallece en modo 'Afterlife'.	1. El personaje muere. 2. El personaje muere nuevamente.	2. El personaje reaparece con el ultimo NPC con el que interactuó en modo 'Afterlife'. 3. El personaje reaparece en el 'respawn'.
FLUJO ALTERNO	ACTOR	SISTEMA
1. El jugador no fallece en modo 'Afterlife' y no encuentra donde falleció.	1. El personaje muere. 2. El personaje sigue jugando en modo 'Afterlife'.	2. El personaje reaparece con el ultimo NPC con el que interactuó en modo 'Afterlife'. 3. El personaje se mantiene en modo 'Afterlife'.
POSTCONDICIÓN	Revivir o Morir	

Table 9: Descripción del caso de uso: Cambiar modo

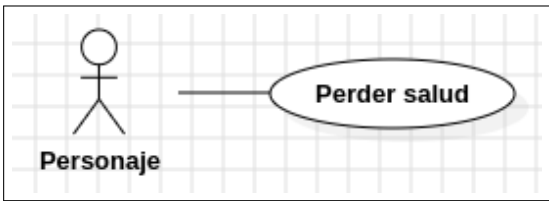


Figure 11: Caso de uso: Perder salud

CASO DE USO	Perder salud	
ACTOR	Personaje (Jugador)	
DESCRIPCIÓN	El jugador pierde puntos de vida al sufrir daño.	
PRECONDICIÓN	Ninguna.	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Choca con un obstáculo.	2. Se disminuye un punto de vida y el personaje regresa al inicio de la habitación.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Vida mínima.	1. Choca con un obstáculo.	2. Fallece y cambia a modo ‘Afterlife’.
POSTCONDICIÓN	Morir	

Table 10: Descripción del caso de uso:

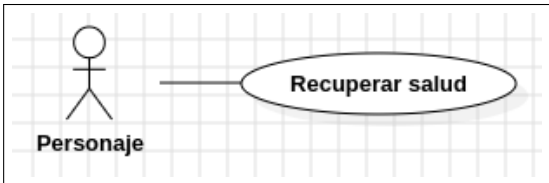


Figure 12: Caso de uso: Recuperar salud

CASO DE USO	Recuperar salud	
ACTOR	Personaje (Jugador)	
DESCRIPCIÓN	Aumenta los puntos de vida de personaje cada que termina una habitación.	
PRECONDICIÓN	Ninguna	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. El jugador termina una habitación.	2. Adquiere un punto de vida.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Vida máxima.	1. El jugador termina una habitación.	2. No recibe puntos adicionales de vida.
POSTCONDICIÓN	Ninguna.	

Table 11: Descripción del caso de uso: Recuperar salud

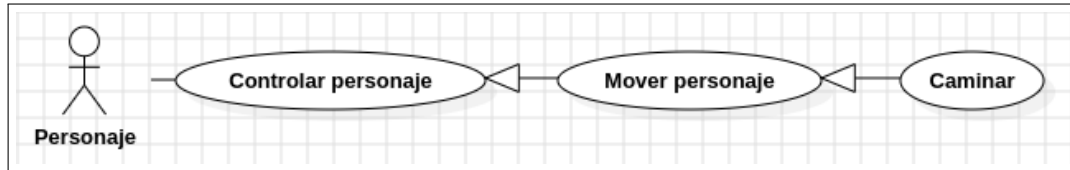


Figure 13: Caso de uso: Caminar

CASO DE USO	Caminar	
ACTOR	Personaje (Jugador)	
DESCRIPCIÓN	El jugador se mueve en el eje horizontal.	
PRECONDICIÓN	Ninguna.	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. El jugador presiona las teclas de movimiento horizontal.	2. El personaje se mueve en la dirección seleccionada.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Colisión.	1. El jugador presiona las teclas de movimiento horizontal. 3. El personaje alcanza un muro.	2. El personaje se mueve en la dirección seleccionada. 4. El personaje colisiona y detiene el movimiento.
POSTCONDICIÓN	Ninguna	

Table 12: Descripción del caso de uso: Caminar

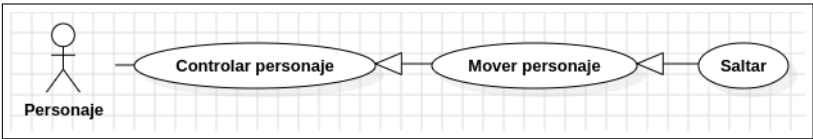


Figure 14: Caso de uso: Saltar

CASO DE USO	Saltar	
ACTOR	Personaje (Jugador)	
DESCRIPCIÓN	El personaje realiza un salto vertical mediante un impulso.	
PRECONDICIÓN	Ninguna.	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. El personaje esta tocando el suelo y presiona la tecla para saltar.	2. Cambio de posición vertical mediante un impulso (curva parabólica).
FLUJO ALTERNO	ACTOR	SISTEMA
1. Colisión.	1. El personaje esta tocando el suelo y presiona la tecla para saltar. 3. El personaje alcanza un muro(techo).	2. Cambio de posición vertical mediante un impulso (curva parabólica). 4. Se cancela la velocidad vertical del personaje para empezar su descenso.
FLUJO ALTERNO	ACTOR	SISTEMA
2. Permision de salto.	1. El personaje se encuentra en el aire. 3. Toca una superficie de piso.	2. Se inhabilita el salto. 4. Se habilita el salto.
POSTCONDICIÓN	Ninguna	

Table 13: Descripción del caso de uso: Saltar

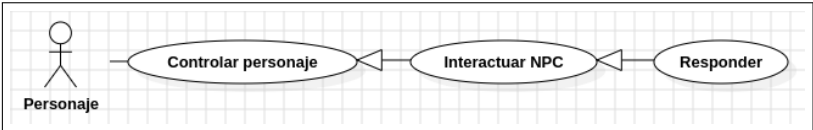


Figure 15: Caso de uso: Responder

CASO DE USO	Responder	
ACTOR	personaje (Jugador)	
DESCRIPCIÓN	El jugador responde el mensaje del NPC desde el teclado.	
PRECONDICIÓN	NPC: mostrar mensaje	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. El jugador escribe el mensaje y lo envía.	2. Se envía el mensaje al sistema inteligente e imprime una respuesta.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Cancelación.	1. El jugador escribe el mensaje y cierra la ventana de dialogo.	2. Cierra la ventana de dialogo y regresa al juego.
POSTCONDICIÓN	Ninguna.	

Table 14: Descripción del caso de uso: Responder

8.4 Diagrama de clases

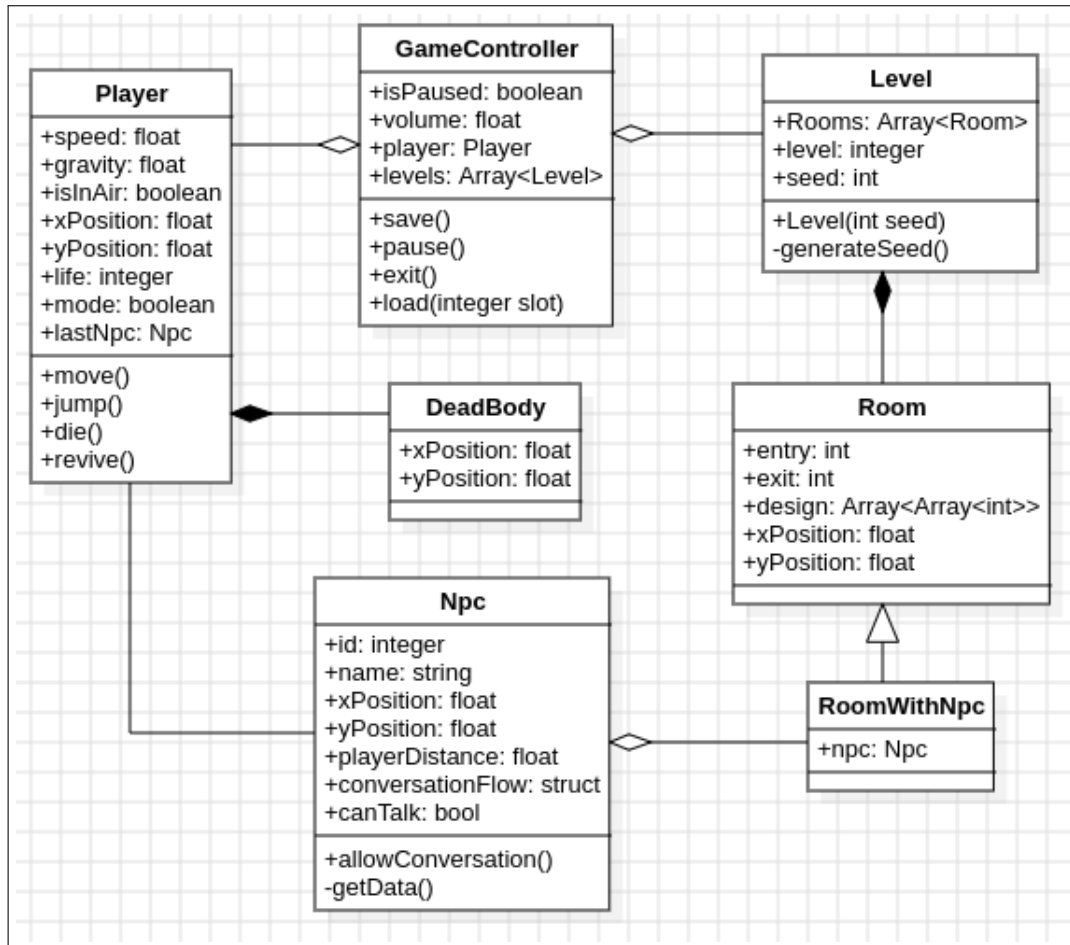


Figure 16: Diagrama de clases

Con base en el genero del videojuego, se requiere de diferentes clases que complementen la funcionalidad lógica. Elementos como la cámara, 'sprites', colisiones, sonidos, partículas, animaciones, cinemáticas, etc. pueden ser añadidas directo desde el IDE de Unity. Por esta razón, el diagrama de clases de la figura 16 omite esos aspectos de construcción (constructores, 'getters' y 'setters' incluidos) y busca representar la estructura y relación entre clases.

8.5 Diagrama de secuencia

8.6 Diagrama de estados

8.7 Diagrama de actividades

9 Prototipo

10 Referencias Bibliográficas

References

- [1] Supergiant Games, *Hades*. Supergiant Games 2019.
- [2] Maddy Makes Games , *Celeste*. Maddy Makes Games, 2018.
- [3] Team Cherry, *Hollow Knight*, Team Cherry, 2017.
- [4] Toge Productions, *Coffee Talk*, Toge Productions, 2020.
- [5] Character Technologies, Inc. “character.ai”, 2022. [En línea]. Disponible: <https://character.ai>. [Accedido: 11-Junio-2024].
- [6] Unity Technologies. “Unity para principiantes — Unity”. Unity. 4. [En línea]. Disponible: <https://unity.com/es/learn/get-started> [Accedido el 13 de junio de 2024].
- [7] E. Canle Fernández. “¿Qué lenguaje de programación usa Unity? — Tokio School”. Tokio School. [En línea]. Disponible: <https://www.tokioschool.com/noticias/lenguaje-unity/#:~:text=C#,%20que%20se%20pronuncia%20'C,sistema%20de%20programacin%20por%20eventos>. [Accedido el 13 de junio de 2024].
- [8] UNAM. “PROLOG ‘”. [En línea]. Disponible: https://virtual.cuautitlan.unam.mx/intar/?page_id=212#:~:text=PROLOG%20utiliza%20un%20lenguaje%20basado,lgica%20partiendo%20de%20predicados%20determinados. [Accedido el 13 de junio de 2024].
- [9] MongoDB. “MongoDB: The Developer Data Platform”. MongoDB. [En línea]. Disponible: <https://www.mongodb.com>. [Accedido el 13 de junio de 2024].
- [10] MKLab. “StarUML”. StarUML. [En línea]. Disponible: <https://www.pixilart.com>. [Accedido el 13 de junio de 2024].
- [11] Pixilart. “Pixilart”. Pixilart LLC. [En línea]. Disponible: <https://staruml.io>. [Accedido el 13 de junio de 2024].