

Plataforma para recreación de estrategia basada en aprendizaje reforzado

Anexo III – Especificación de análisis y del diseño

Grado en Ingeniería Informática



**VNiVERSiDAD
D SALAMANCA**

Julio de 2023

Autor

Erick José Mercado Hernández

Tutor/a

Vidal Moreno Rodilla

Lista de cambios

Número	Fecha	Versión	Autor
0	02-02-2023	Versión 0.1(Modelo de requisitos inicial)	Erick José Mercado Hernández
1	01-07-2023	Versión 1.0 (Desarrollo del documento)	Erick José Mercado Hernández

Índice de contenido

1.- Introducción	1
2.- Ámbito de software	2
3.- Diseño arquitectónico.....	2
3.1.- División por paquetes.....	2
3.2.- Patrones utilizados	3
3.2.1.- Patrón Singleton	3
3.2.2.- Patrón Abstract Factory.....	4
3.3.- Diagrama de clases de diseño.....	4
4.- Diseño de la interfaz y escenarios	7
5.- Diseño procedimental.....	12
5.1.- Realización de casos de análisis.....	12
6.- Plan de desarrollo, implementación y despliegue	22
6.1.- Plan de implementación	22
6.2.- Plan de despliegue.....	22
7.- Conclusiones	23

Índice de Tablas

No se encuentran elementos de tabla de ilustraciones.

Índice de ilustraciones

Ilustración 1: Diagrama de paquetes	2
Ilustración 2: Pseudocódigo del patrón Singleton	3
Ilustración 3: Representación del patrón de diseño usado en el proyecto	4
Ilustración 4: Diagrama de clases (I)	4
Ilustración 5: Diagrama de clases (II)	5
Ilustración 6: Diagrama de clases (III)	5
Ilustración 7: Diagrama de clases (IV)	6
Ilustración 8: Panel de Victoria/Derrota	7
Ilustración 9: Panel de Pausa	8
Ilustración 10: UI del turno de la IA	8
Ilustración 11: Panel Menú de Inicio	9
Ilustración 12: Panel de selección de Escenario	9
Ilustración 13: Interfaz de acciones y retrato de la unidad	10
Ilustración 14: Escenario (I) Ruinas	10
Ilustración 15: Escenario (II) El Pueblo	11
Ilustración 16: Diagrama de análisis del caso de uso "Menú Principal"	12
Ilustración 17: Diagrama de análisis del caso de uso "Elegir Escenario"	13
Ilustración 18: Diagrama de análisis del caso de uso "Seleccionar Unidad"	13
Ilustración 19: Diagrama de análisis del caso de uso "Mover Unidad"	14
Ilustración 20: Diagrama de análisis del caso de uso "Atacar cuerpo a cuerpo"	14
Ilustración 21: Diagrama de análisis del caso de uso "Atacar a distancia"	15
Ilustración 22: Diagrama de análisis del caso de uso "Curar Aliado"	15
Ilustración 23: Diagrama de análisis del caso de uso "Atacar con magia"	16
Ilustración 24: Diagrama de análisis del caso de uso "Finalizar turno"	16
Ilustración 25: Diagrama de análisis del caso de uso "Configurar opciones del juego"	17
Ilustración 26: Diagrama de análisis del caso de uso "Ganar/Perder Escenario"	17
Ilustración 27: Diagrama de análisis del caso de uso "Reiniciar Escenario"	18
Ilustración 28: Diagrama de análisis del caso de uso "Salir del juego"	18
Ilustración 29: Diagrama de análisis del caso de uso "Pausar juego"	19
Ilustración 30: Diagrama de análisis del caso de uso "Guardar Partida"	19
Ilustración 31: Diagrama de análisis del caso de uso "Cargar Partida"	20
Ilustración 32: Diagrama de análisis del caso de uso "Consultar tutorial"	20
Ilustración 33: Diagrama de análisis del caso de uso "Interactuar escenario"	21
Ilustración 34: Diagrama de análisis del caso de uso "Destruir Objeto"	21

Ilustración 35: Diagrama de despliegue	22
--	----

1.- Introducción

En el siguiente documento se va a presentar la especificación del análisis y del diseño para el presente proyecto.

En el documento se va a seguir la siguiente estructura. Primero, se van a presentar los objetivos a conseguir y posteriormente, se van a mostrar el diseño arquitectónico, el modelo de análisis del sistema y el modelo de diseño.

2.- Ámbito de software

El sistema va a tratar de una plataforma de desarrollo de estrategia, que mediante juego TBRPG, que implementa un algoritmo de aprendizaje reforzado.

El principal objetivo del proyecto va a consistir en la creación de una IA funcional que sea desafiante estratégicamente contra el jugador, así como ofrecer otras funcionalidades extra que pueden ser de utilidad.

3.- Diseño arquitectónico

En este apartado se va a mostrar la arquitectura del proyecto y los patrones utilizados tanto en el diseño como en la arquitectura.

Se va a comenzar por la división en paquetes, posteriormente el

3.1.- División por paquetes

Se divide el sistema en los siguientes paquetes que se mostraran a continuación. Cada uno se va a corresponder con una tarea de alto nivel que va a realizar el sistema:

- **Unity Engine:** Motor grafico que permite la realización de todos los paquetes siguientes y que contiene toda aquella lógica que permite la creación y ejecución del juego.
- **Gestión de unidades:** Contiene toda aquella lógica que permite a los personajes interactuar con el entorno.
- **Gestión de datos:** Contiene la lógica que permite guardar y cargar los datos de una partida, así como las preferencias del jugador.
- **Gestión principal:** Contiene toda la lógica de nuestro juego, desde el sistema de turnos, paneles, etc.

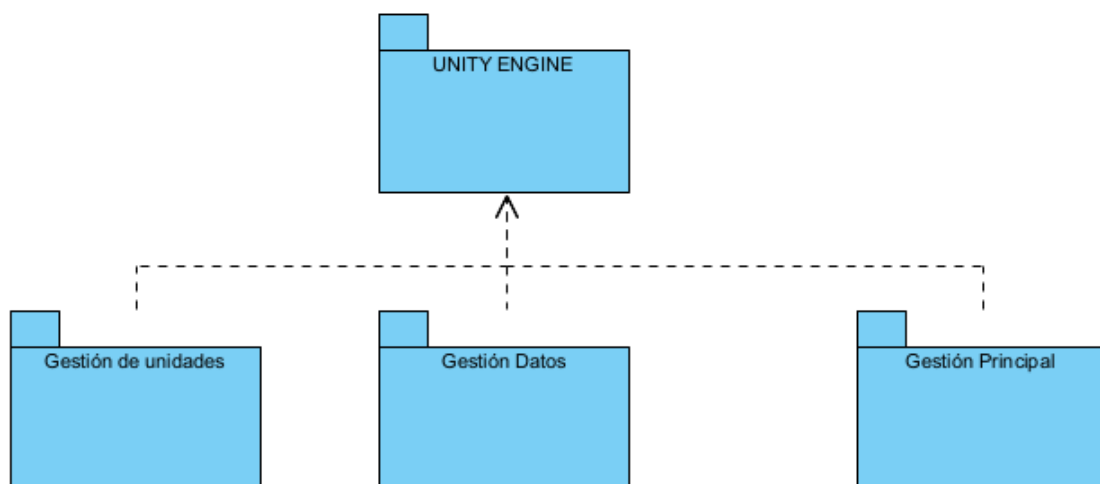


Ilustración 1: Diagrama de paquetes

3.2.- Patrones utilizados

En esta sección se van a introducir los diferentes patrones que van a tener un peso en el desarrollo del proyecto.

3.2.1.- Patrón Singleton

El patrón Singleton es uno de los patrones de diseño más utilizados en la programación orientada a objetos. Su principal objetivo es garantizar que solo pueda existir una única instancia de una clase en todo el sistema. Este patrón se utiliza comúnmente para representar los servicios globales, recursos compartidos y otros componentes que no deben tener múltiples instancias.

Se compone de una única clase que es responsable de crear su propia instancia única. Esta clase tiene un constructor privado para evitar que se instancie externamente. También proporciona un método estático que controla el acceso a la única instancia de la clase.

```
clase Singleton
    privado estático instancia de Singleton

    // Constructor privado para prevenir instanciación
    privado constructor()
        // Nada aquí porque este es un constructor vacío

    // Método estático para obtener la única instancia
    público estático función obtenerInstancia()
        // Comprobar si no existe la instancia
        si instancia es igual a nulo entonces
            // Si no existe, crear una nueva
            instancia = nuevo Singleton()
        fin si
        // Devolver la instancia única
        retornar instancia
    fin función
fin clase
```

Ilustración 2: Pseudocódigo del patrón Singleton

El patrón Singleton tiene varias ventajas, como el control estricto sobre las instancias de la clase, la garantía de un punto de acceso global y la posibilidad de retrasar la inicialización de recursos hasta que se necesiten.

Sin embargo, el patrón Singleton también tiene algunas desventajas. La principal es que puede ser difícil de implementar correctamente en entornos de programación multihilo. También puede dificultar la escritura de pruebas unitarias debido a su estado global. Adicionalmente, puede fomentar el uso de un modelo de programación basado en estados globales, lo que puede llevar a un código más difícil de entender y mantener.

3.2.2.- Patrón Abstract Factory

El patrón de diseño Abstract Factory, o Fábrica Abstracta, es un patrón creacional que proporciona una interfaz para crear familias de objetos relacionados sin especificar sus clases concretas. En otras palabras, este patrón permite a los clientes utilizar los métodos proporcionados para crear diferentes objetos de diferentes familias sin tener que preocuparse por los detalles de implementación de estas clases.

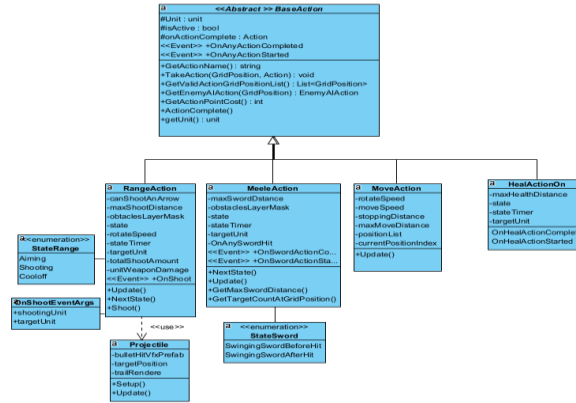


Ilustración 3: Representación del patrón de diseño usado en el proyecto

3.3.- Diagrama de clases de diseño

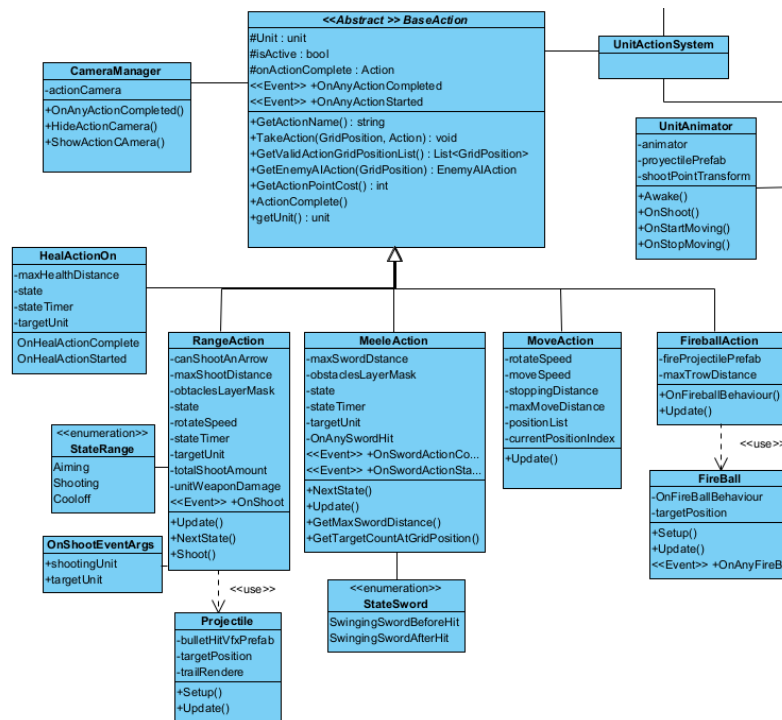


Ilustración 4: Diagrama de clases (I)

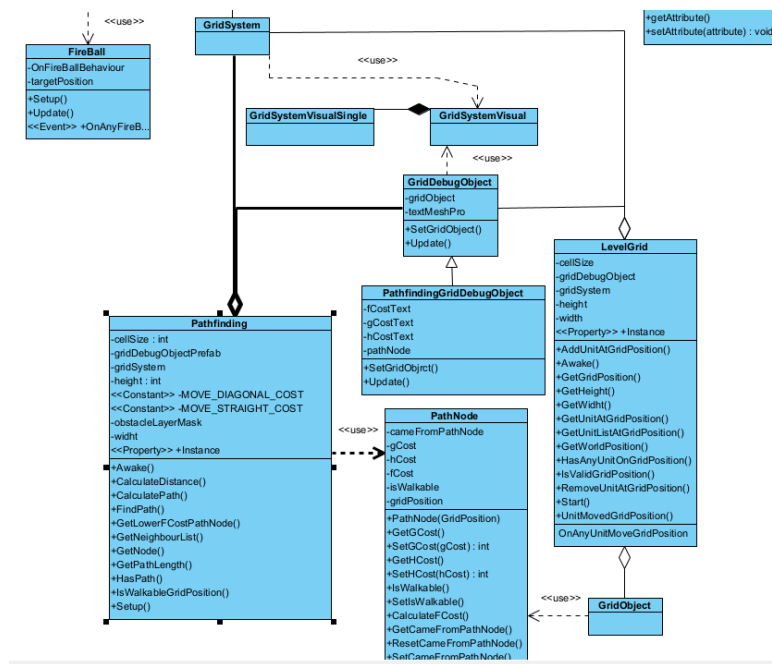


Ilustración 5: Diagrama de clases (II)

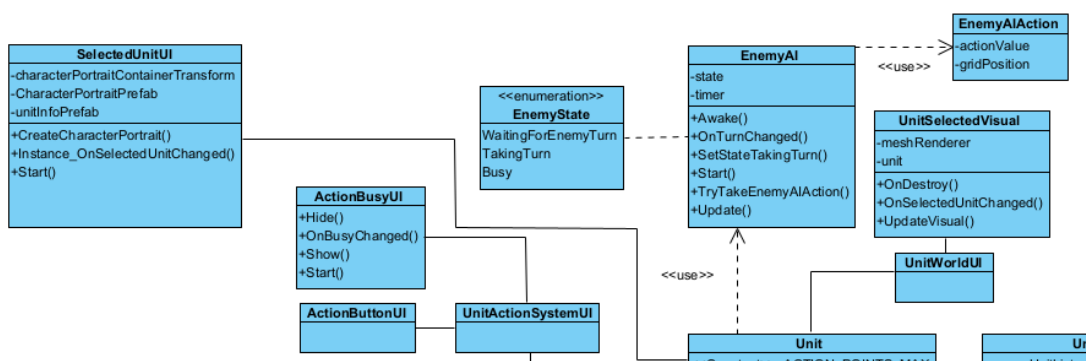


Ilustración 6: Diagrama de clases (III)

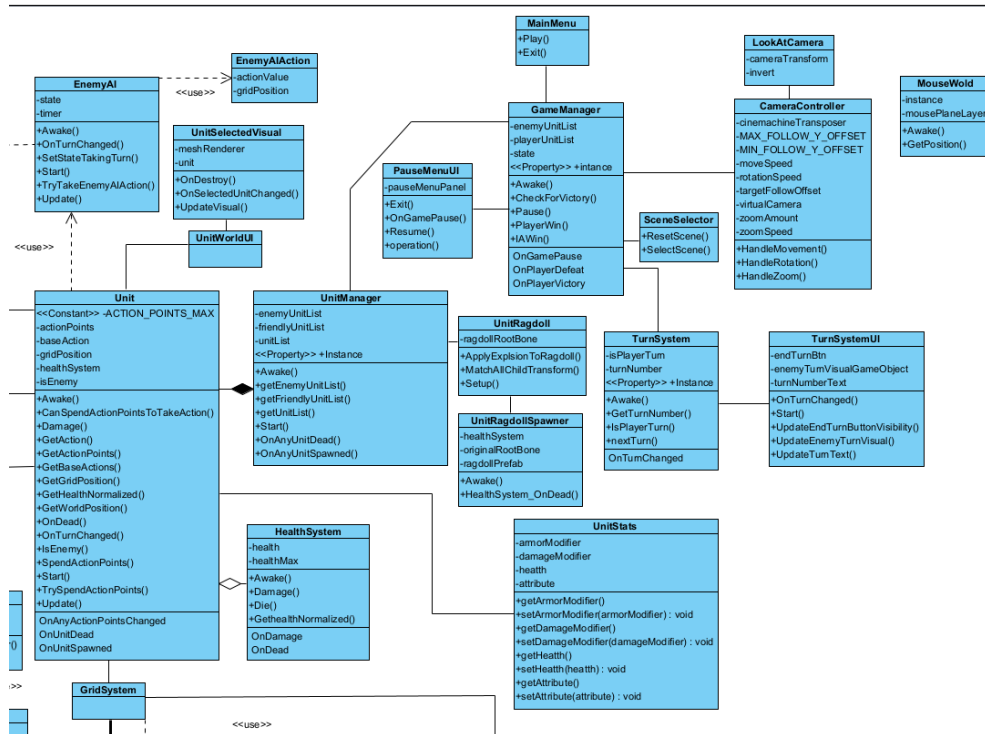


Ilustración 7: Diagrama de clases (IV)

4.- Diseño de la interfaz y escenarios

En este apartado se mostrarán el diseño de las interfaces ofrecidas por los componentes del sistema. Más en concreto se mostrará las diferentes visualizaciones de los paneles y escenarios que se han creado para el proyecto.

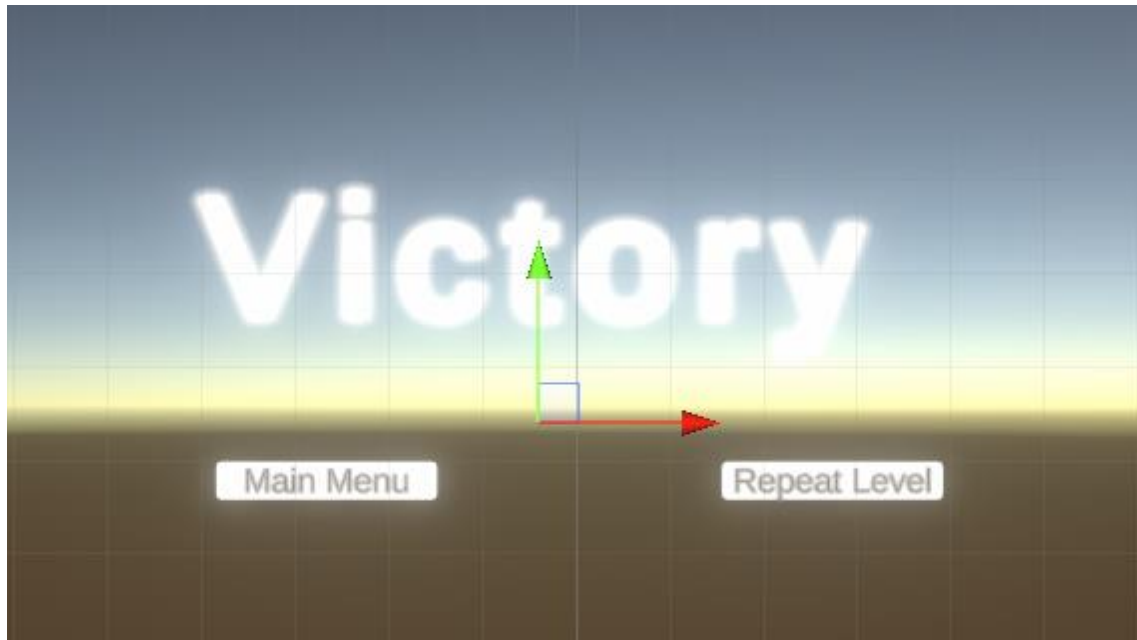


Ilustración 8: Panel de Victoria/Derrota

En la ilustración 8 se muestra como se ve el panel de victoria, se ha preferido dejar una interfaz simple donde se muestren dos botones principales, en color blanco y letras negras, acorde con el estilo “Low Polygons” del juego.

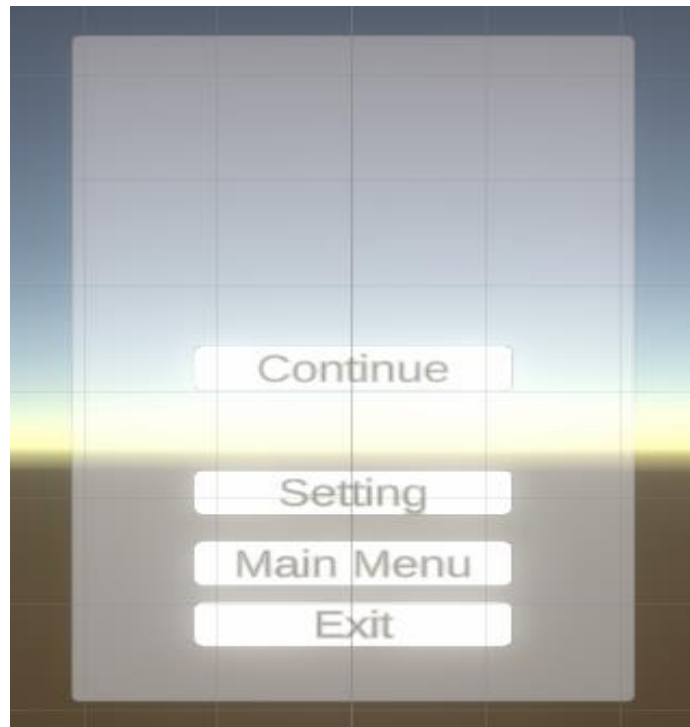


Ilustración 9: Panel de Pausa

En la ilustración 9 se muestra el menú de pausa que sigue exactamente el mismo patrón del menú anterior, letras negras sobre fondo blanco minimalista. Además el fondo se muestra en un blanco difuminado para que contraste mejor cuando accedemos a este menú.



Ilustración 10: UI del turno de la IA

En la ilustración 10 se muestra el panel que bloquea las acciones del jugador cuando es el turno del enemigo, se ha utilizado un fondo negro y letras de un color naranja neón para que pueda verse correctamente en todos los escenarios del juego y oculte correctamente los botones de acción de las unidades enemigas.

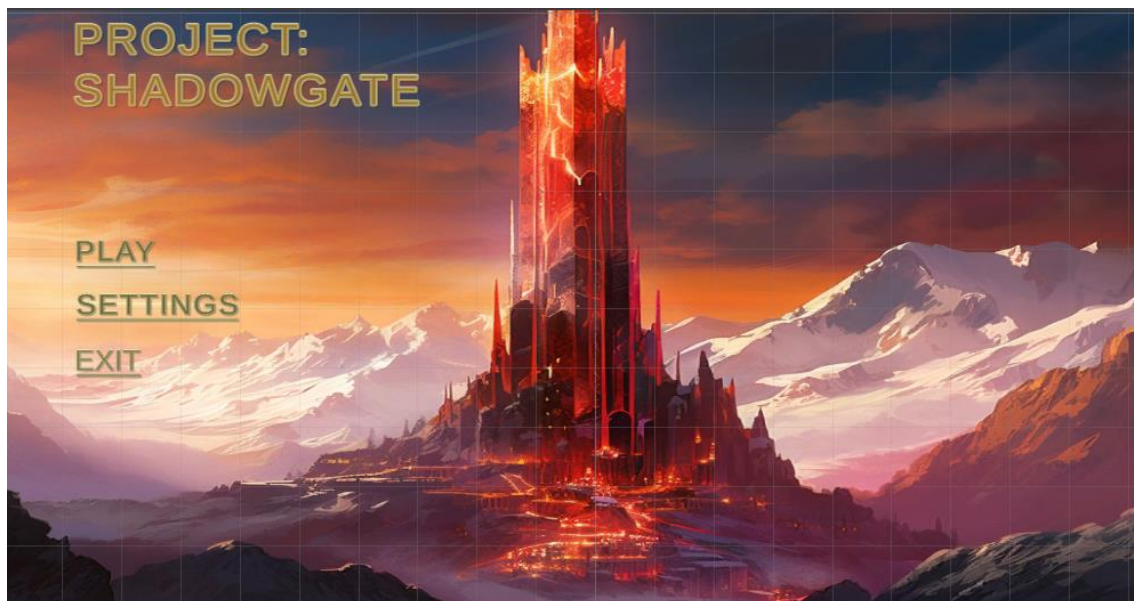


Ilustración 11: Panel Menú de Inicio

En este panel se muestra el menú de inicio, para el fondo se ha seleccionado una imagen generada por una inteligencia artificial y que corresponde muy bien con el nombre en clave que se le ha dado al proyecto, esta muestra una enorme ciudadela con los colores del ocaso rodeada de una cordillera de montañas nevadas. El texto elegido es de color dorado para que contraste de forma correcta con el fondo y sea distinguible para los jugadores. Además, cada botón que se muestra tiene un hover en negro que nos indica si ese botón está seleccionado o no.



Ilustración 12: Panel de selección de Escenario

En la ilustración 12 se muestra un mapa inspirado en la unión de las Islas Canarias y que sirve de paso intermedio para que podamos seleccionar que escenario vamos a jugar.

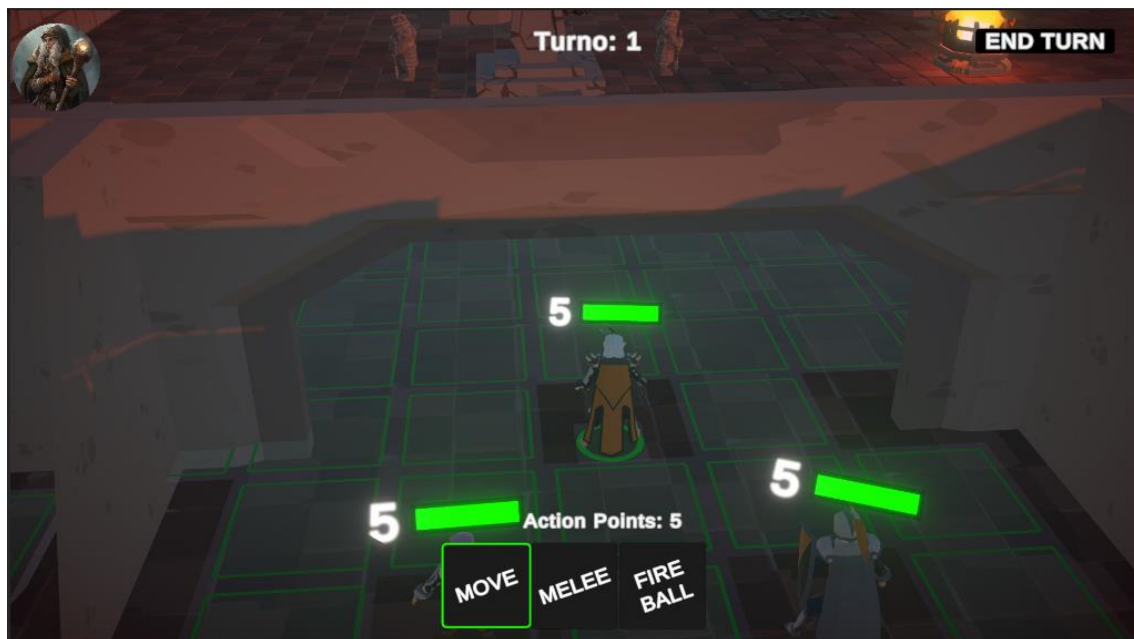


Ilustración 13: Interfaz de acciones y retrato de la unidad

En esta interfaz ya nos encontramos dentro del juego y se muestra una unidad seleccionada. Para los recuadros que puede moverse se han pintado de verde, asumiendo que son los puntos alcanzables dentro del radio de la unidad. Se muestra también sobre la cabeza de las unidades los puntos de acción que puede realizar y la vida restante en forma de barra.

Podemos ver cuando seleccionamos a la unidad que se muestran los botones con las acciones que puede realizar, se utiliza un fondo negro con letras en blanco encima y un hover verde para indicar que el botón está seleccionado. También se muestra el retrato de la unidad en cuestión.

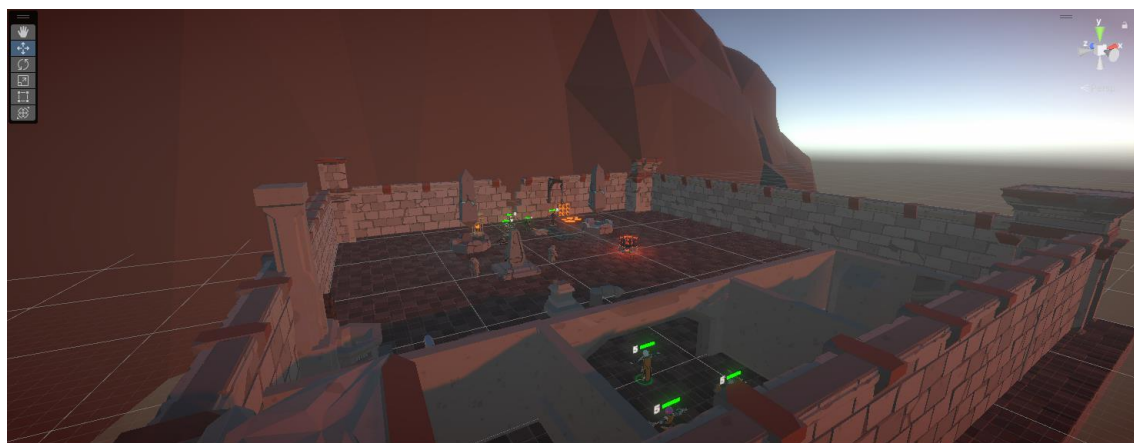


Ilustración 14: Escenario (I) Ruinas

En la ilustración 14 se muestra el primer escenario, llamado “The Walls” como se vio en la imagen del mapa, está inspirado en unas ruinas enanas al borde de una gran montaña. Los colores que predominan aquí son colores cálidos, simulando el sol de un amanecer, los grises de las paredes y monumentos y el naranja cálido del fuego.

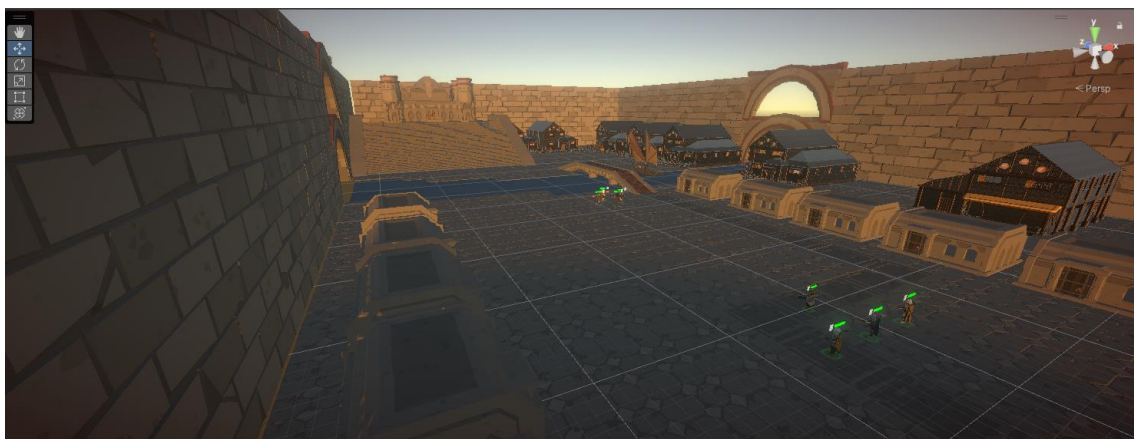


Ilustración 15: Escenario (II) El Pueblo

En esta última ilustración se pretende mostrar el mapa conocido como “Town” en el menú de selección anterior, en este mapa priman los colores fríos del atardecer sobre un pueblo amurallado, el azul del río y el gris de los edificios principales y el pavimento.

5.- Diseño procedimental

A continuación, se va a describir la programación del sistema. Para ello se va a seguir una estructura que se enfoca en facilitar la comprensión de los procesos a implementar en el sistema, ya que estos se presentan primero a un alto nivel mediante análisis de diagramas de secuencia.

5.1.- Realización de casos de análisis

Se va a presentar el diseño de la programación que se corresponde con el análisis de la aplicación exponiendo el funcionamiento del sistema a un alto nivel.

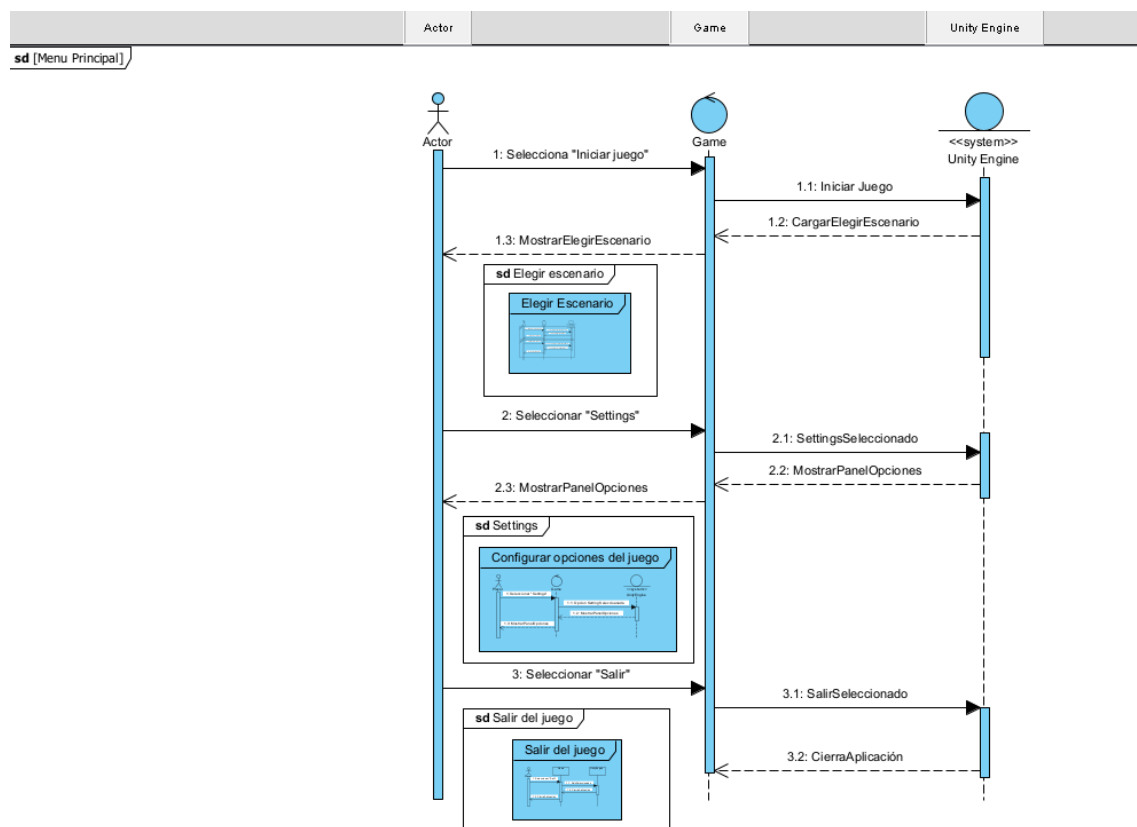


Ilustración 16: Diagrama de análisis del caso de uso "Menú Principal"

En la ilustración 16 se muestra el primer diagrama de análisis donde se ve como el jugador puede iniciar seleccionar diferentes acciones, entre ellas se encuentran iniciar juego, ver el menú de opciones o salir del juego.

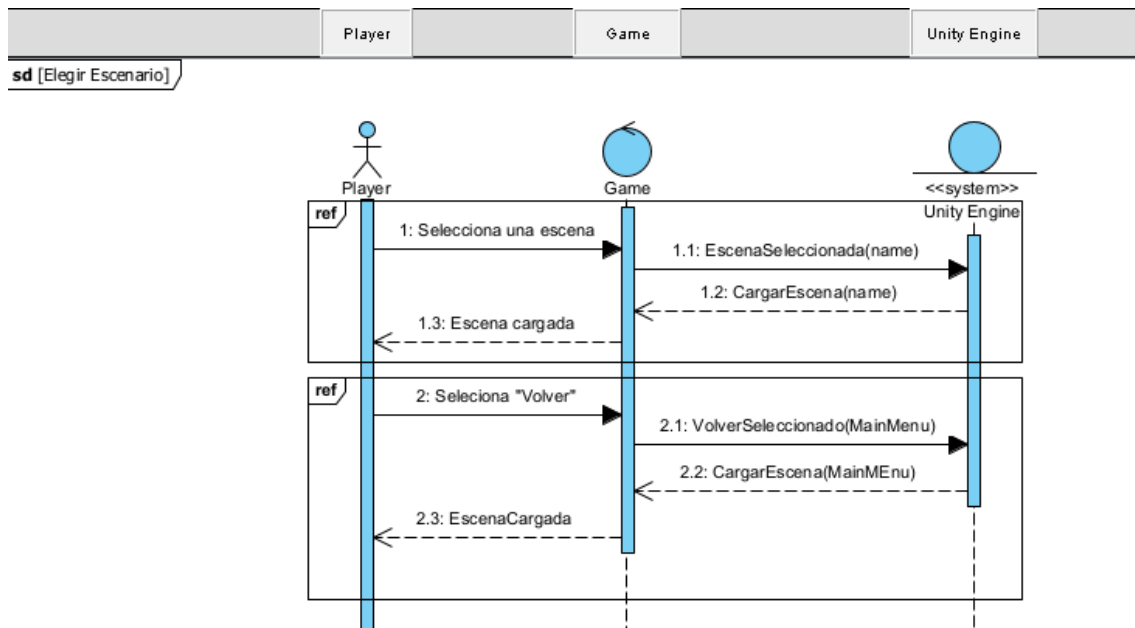


Ilustración 17: Diagrama de análisis del caso de uso "Elegir Escenario"

Una vez seleccionado en el diagrama del caso anterior “iniciar juego” se nos muestra la escena “elegir escenario” en este caso podemos elegir los escenarios jugables a través de un botón, también se puede volver atrás pulsando el botón “volver”.

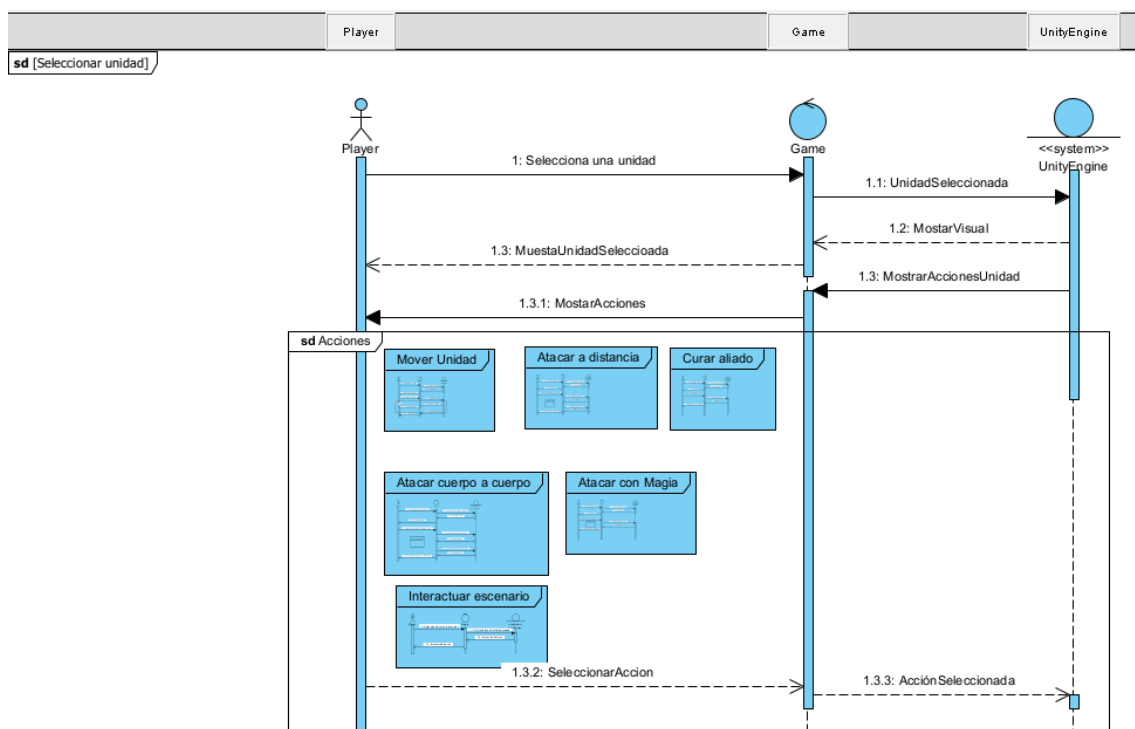


Ilustración 18: Diagrama de análisis del caso de uso "Seleccionar Unidad"

Para elegir una unidad solo tenemos que seleccionar con el ratón, pasamos por encima del collider de la unidad y una vez seleccionada se nos mostraran las acciones que esta unidad posee.

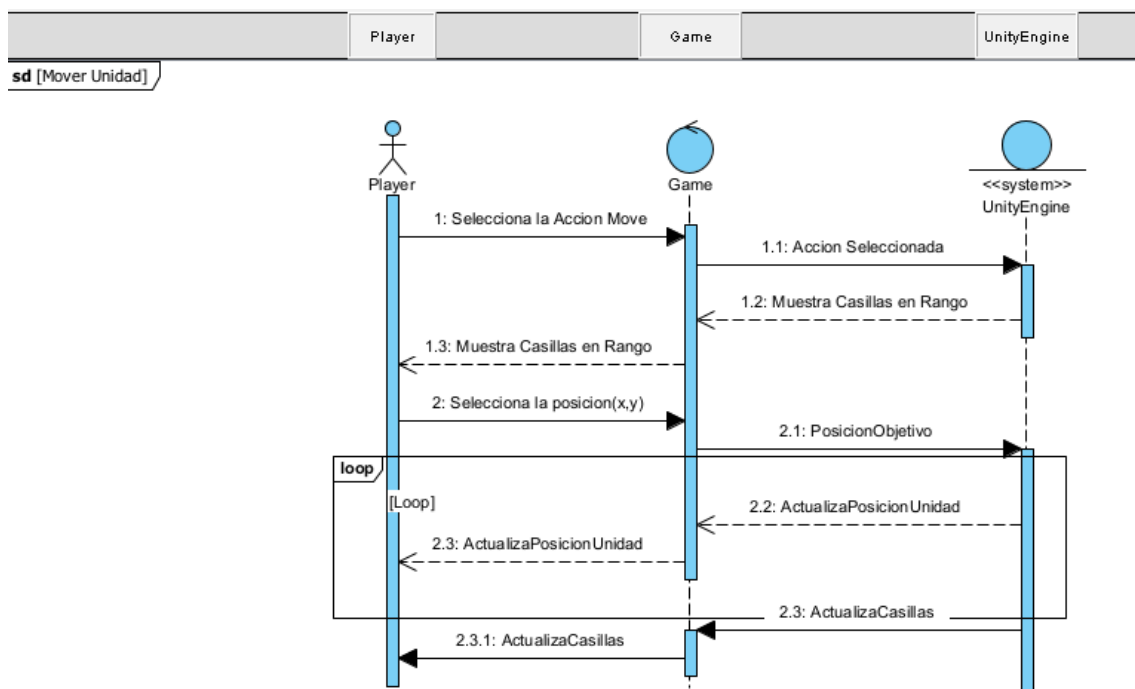


Ilustración 19: Diagrama de análisis del caso de uso "Mover Unidad"

Para mover a la unidad el jugador tiene que seleccionar la acción mover, con la cual nos mostrará un área en color verde donde la unidad se puede mover, luego selecciona una casilla dentro de esa área y la unidad ira actualizando la posición en cada casilla que pise hasta llegar al objetivo marcado.

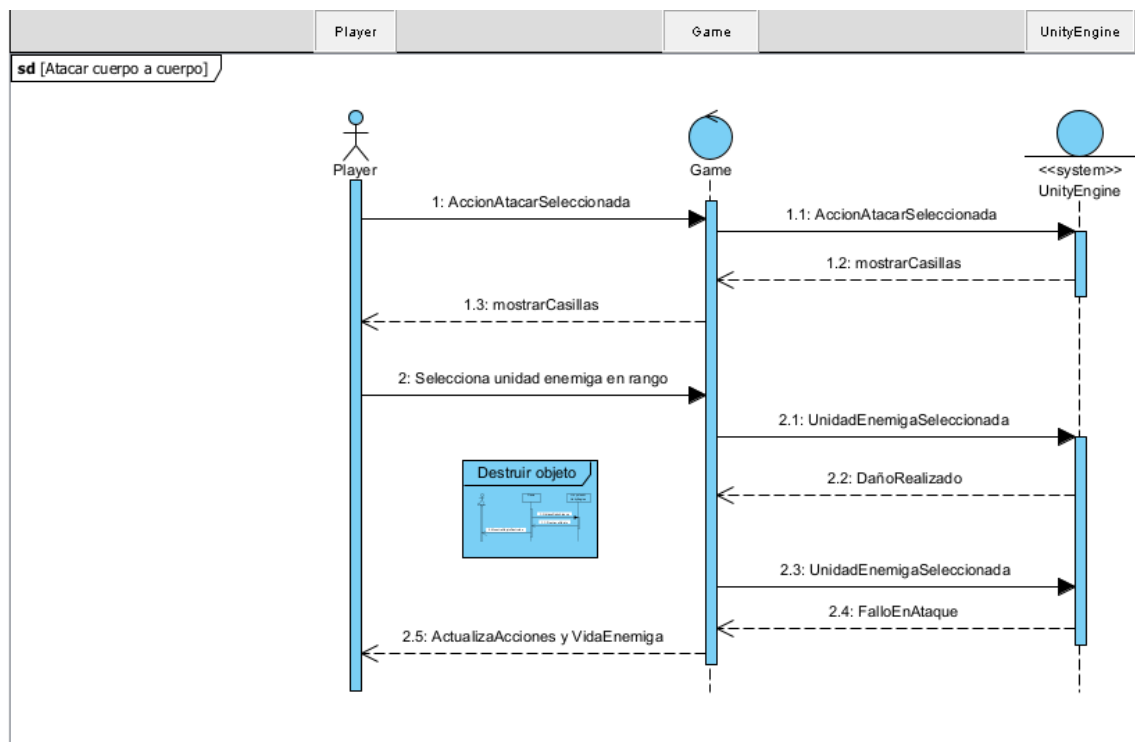


Ilustración 20: Diagrama de análisis del caso de uso "Atacar cuerpo a cuerpo"

Para realizar un ataque cuerpo a cuerpo la unidad debe tener una unidad enemiga en su rango para poder realizar la acción. Si tenemos una unidad o objeto destruible dentro de nuestro rango de acción, podemos realizar el ataque dañando al objetivo.

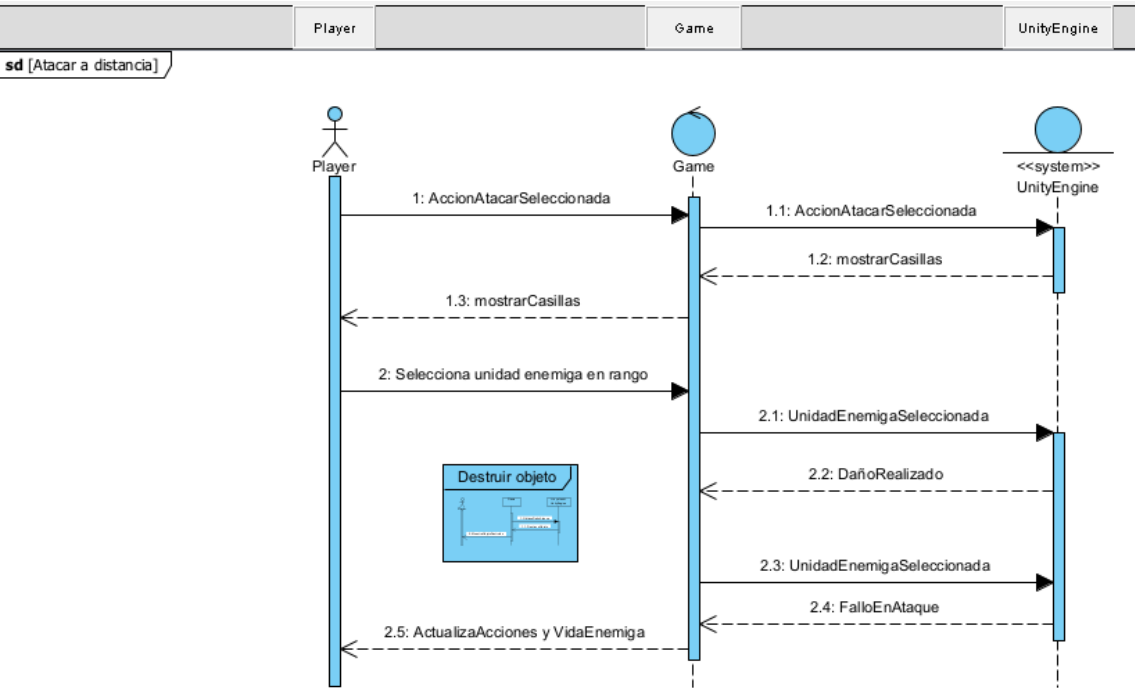


Ilustración 21: Diagrama de análisis del caso de uso "Atacar a distancia".

En este caso de uso sigue el mismo patrón que el caso de uso anterior, solo podemos realizar un ataque a distancia si el enemigo o un objeto destruible se encuentra dentro de nuestro rango de acción.

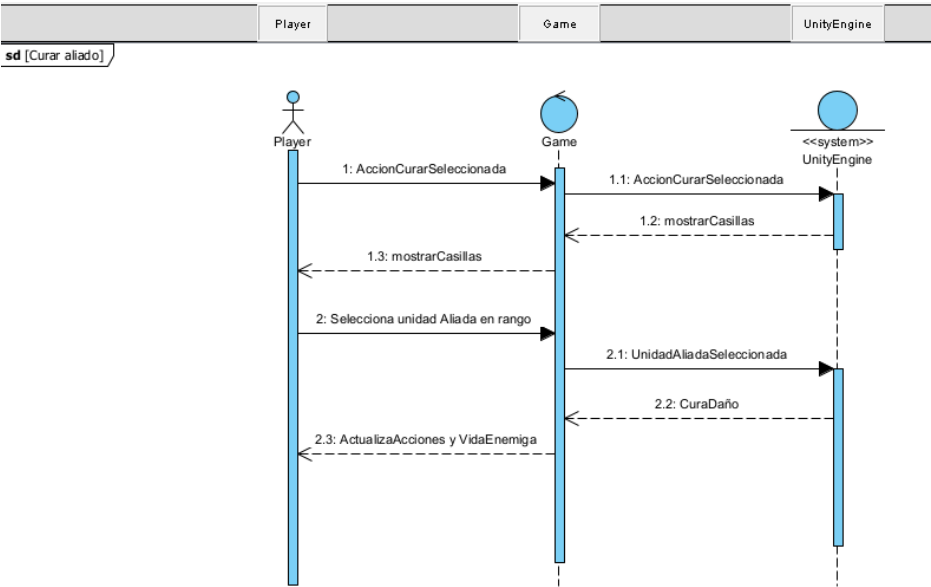


Ilustración 22: Diagrama de análisis del caso de uso "Curar Aliado"

Este caso de uso de la ilustración 22 nos muestra la secuencia de como curar a un aliado, sigue el mismo patrón que el caso de uso anterior de atacar solo que cambiando el objetivo disponible por los aliados.

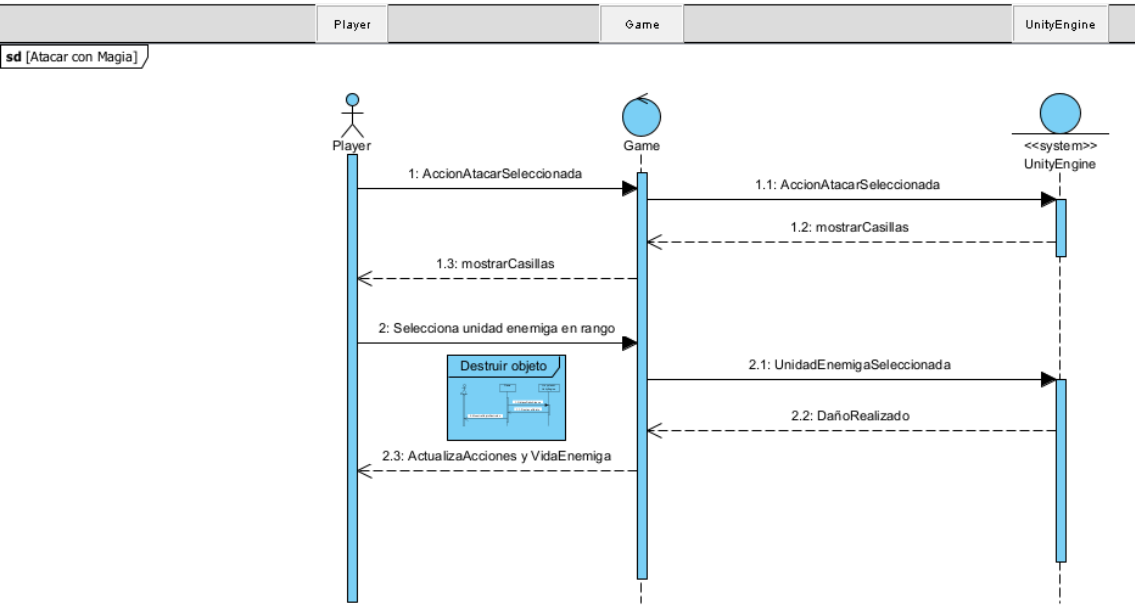


Ilustración 23: Diagrama de análisis del caso de uso "Atacar con magia"

En el caso de uso “Atacar con magia” que se muestra en la ilustración 23 se puede ver que sigue la misma secuencia de los casos de uso de ataque anteriores, además este caso tiene como efecto generar daño en un área, la cual puede dañar a las unidades aliadas.

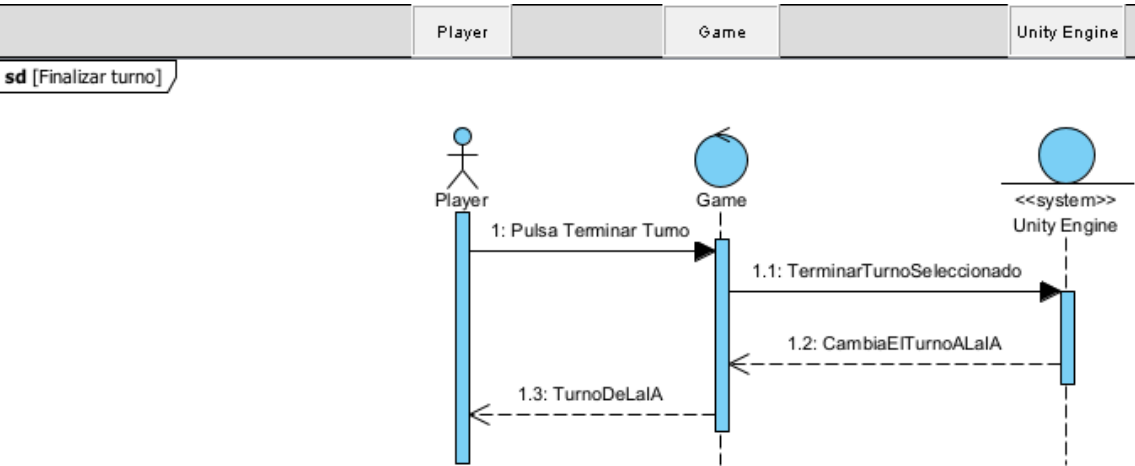


Ilustración 24: Diagrama de análisis del caso de uso "Finalizar turno"

En la ilustración 24 se muestra como pasar turno, existen diferentes maneras de pasar turno, una es pulsando sobre el botón terminar turno, otra es pulsando sobre la tecla

espacio y la última y la que usara la IA es que todas las unidades terminen sus puntos de acción.

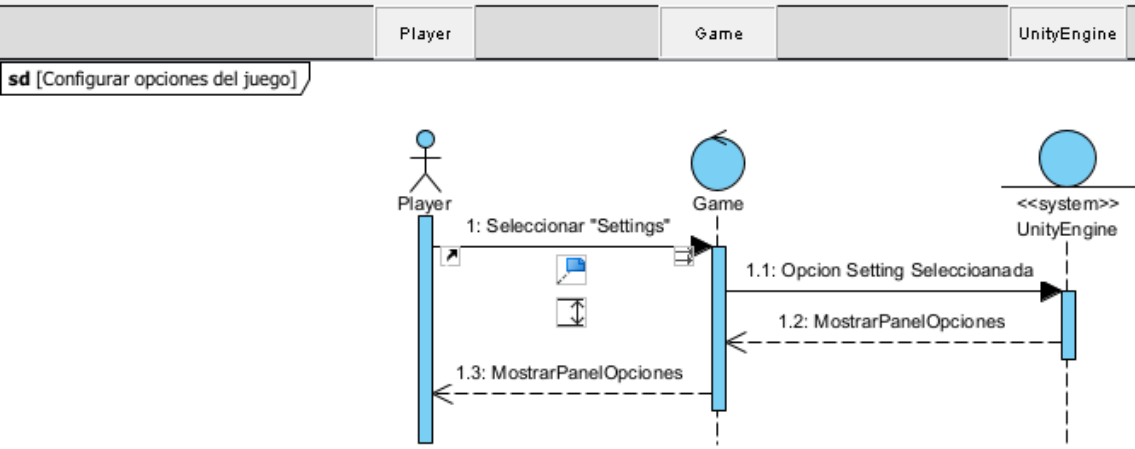


Ilustración 25: Diagrama de análisis del caso de uso "Configurar opciones del juego"

En la ilustración 16 se muestra como podemos acceder desde el menú de inicio a las opciones que puede modificar el jugador, la secuencia seria seleccionar el botón "Settings" dentro del menú de inicio, aunque también podemos encontrarlo dentro del menú de pausa durante el juego.

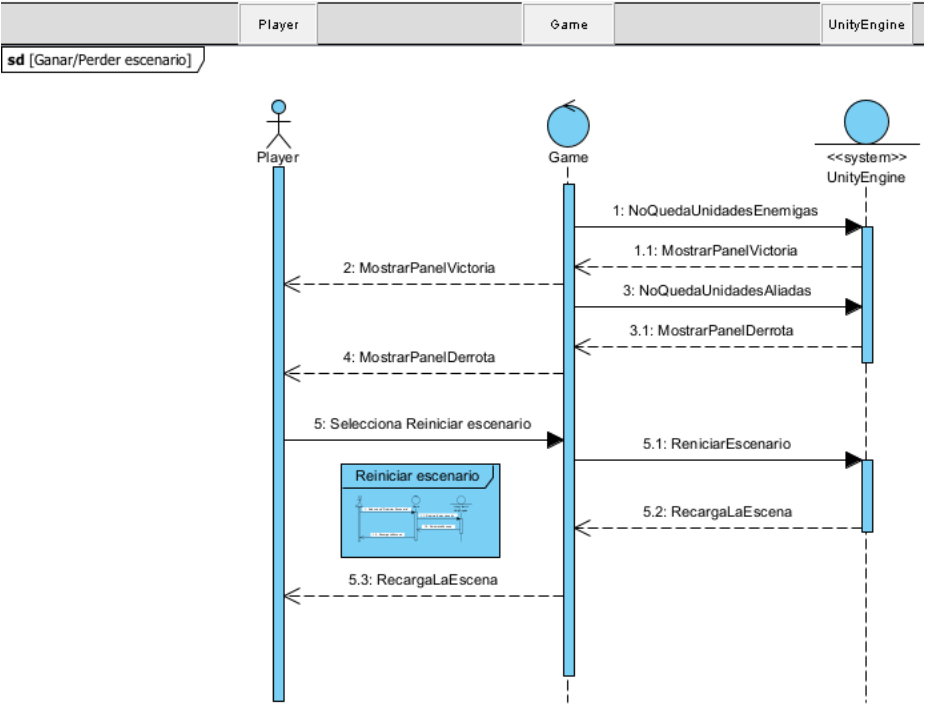


Ilustración 26: Diagrama de análisis del caso de uso "Ganar/Perder Escenario"

La secuencia que sigue este caso de uso se ejecuta cuando dentro de la lista de unidades enemigas o aliadas no queda ningún elemento, mostrando este por pantalla el mensaje de

victoria o derrota, además de las opciones de recargar el escenario o de volver al menú de inicio.

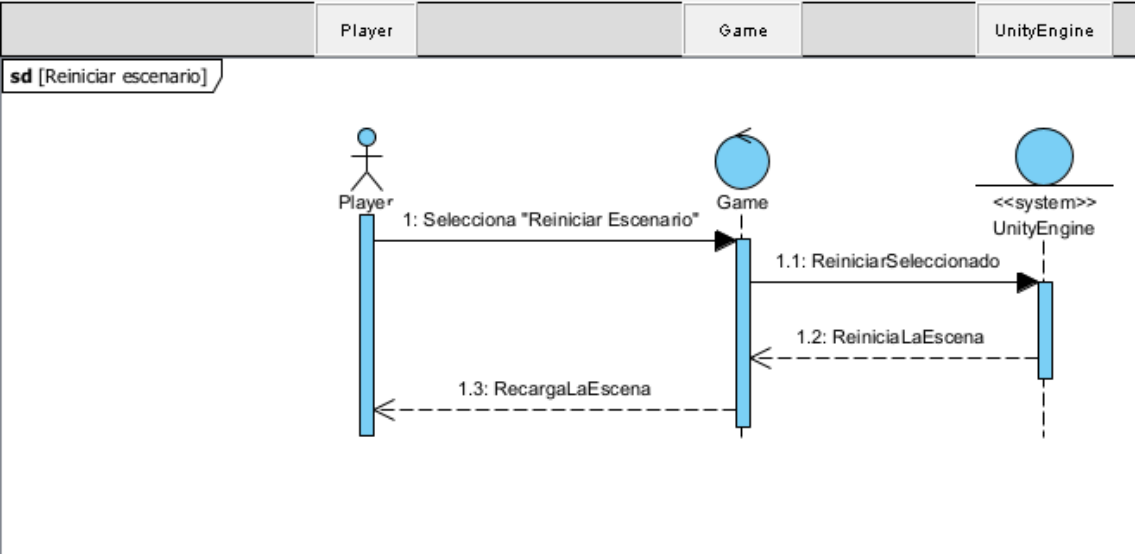


Ilustración 27: Diagrama de análisis del caso de uso "Reiniciar Escenario"

Esta opción se nos muestra cuando ganamos o perdemos una partida, si pulsamos el botón podemos se recargará el escenario a su etapa inicial.

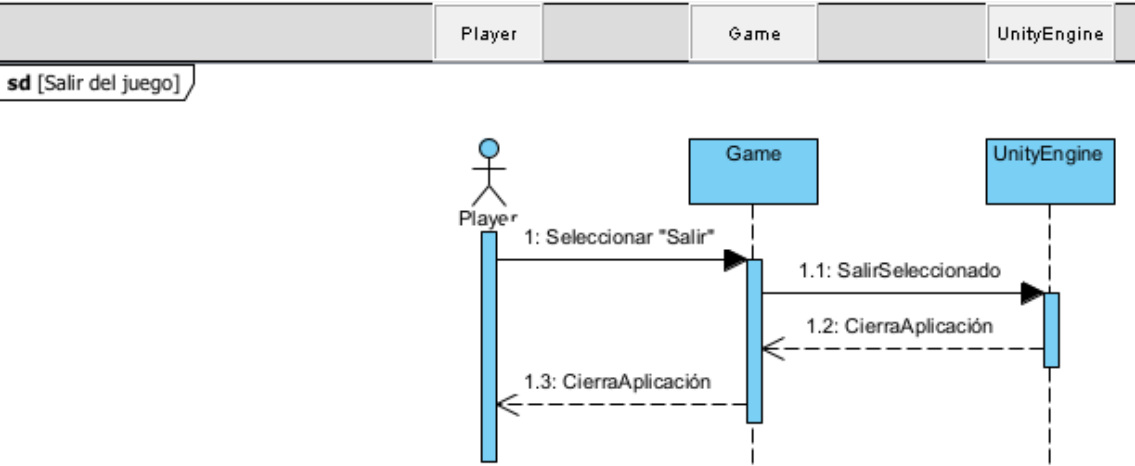


Ilustración 28: Diagrama de análisis del caso de uso "Salir del juego"

Cuando seleccionamos el botón de salir, el juego se cerrara automáticamente, podremos acceder a este caso de uso tanto desde el menú de inicio como desde el menú de pausa.

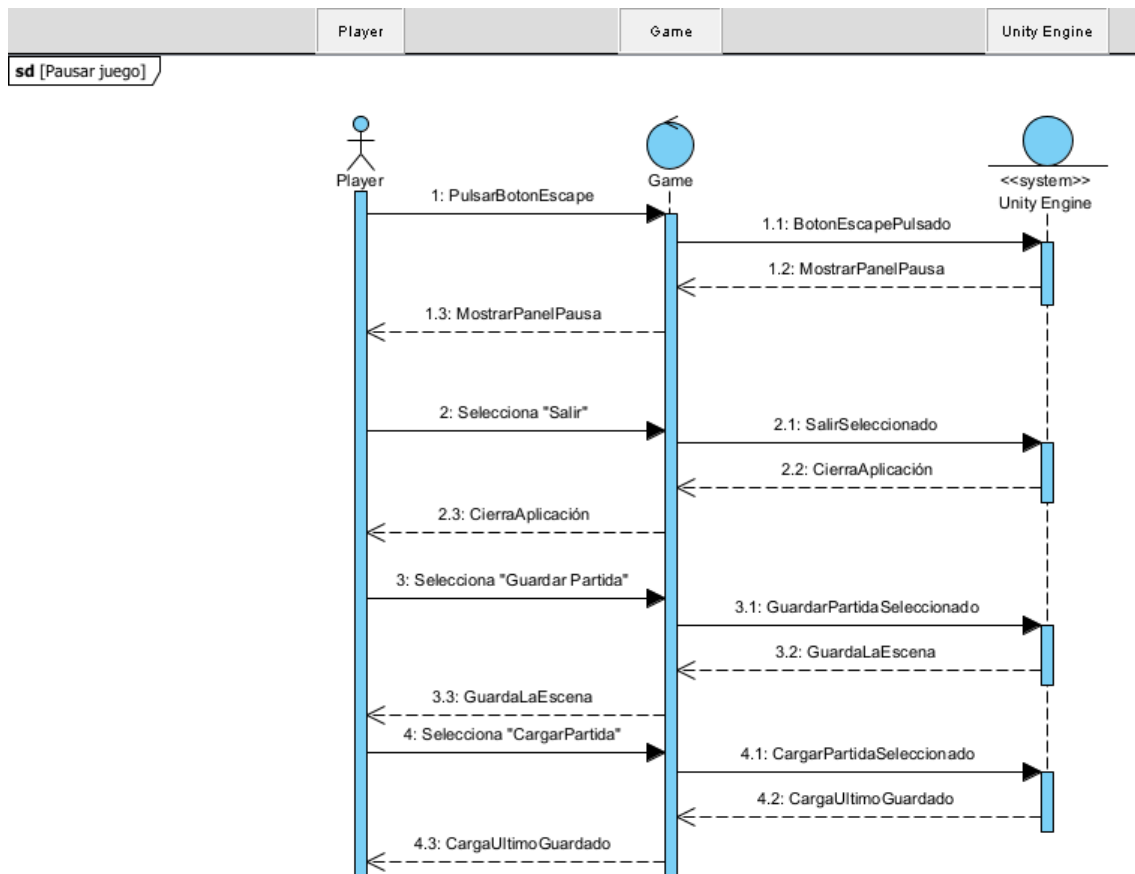


Ilustración 29: Diagrama de análisis del caso de uso "Pausar juego"

Para ejecutar el caso de uso de la ilustración 29, durante nuestra partida solo tendremos que pulsar sobre la tecla "escape" se nos desplegara los casos de uso como salir, guardar, cargar o seleccionar las opciones.

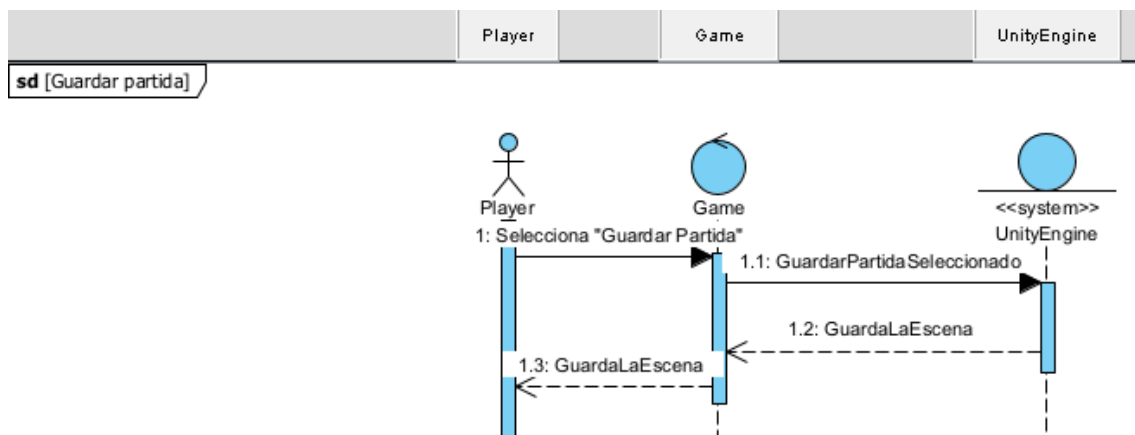


Ilustración 30: Diagrama de análisis del caso de uso "Guardar Partida"

En este caso de uso si hemos activado el menú de pausa dentro del juego podemos seleccionar la opción guardar, que guardara el estado actual de la partida en un archivo binario.

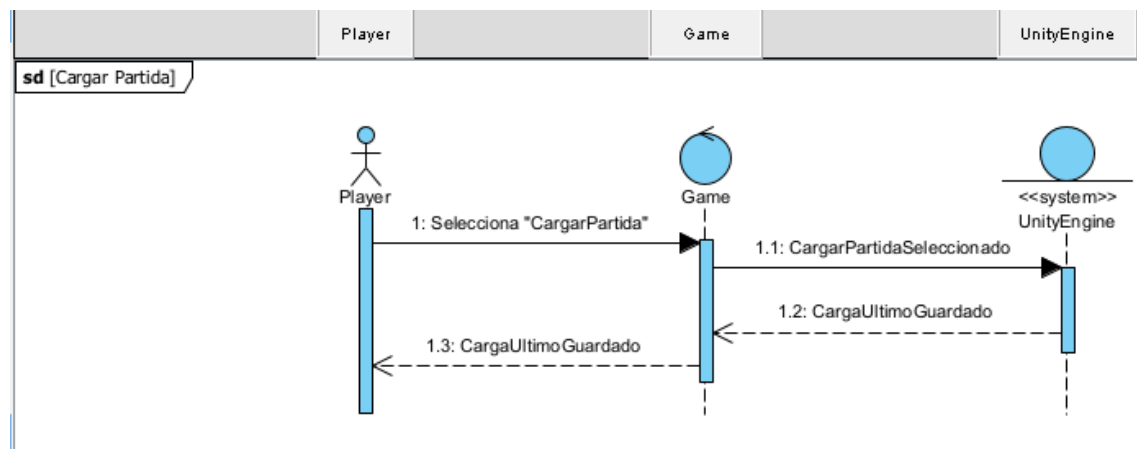


Ilustración 31: Diagrama de análisis del caso de uso "Cargar Partida"

En este caso de uso podemos acceder desde la pantalla principal, cojera el último archivo guardado y cargara la última escena en el estado que fue guardada.

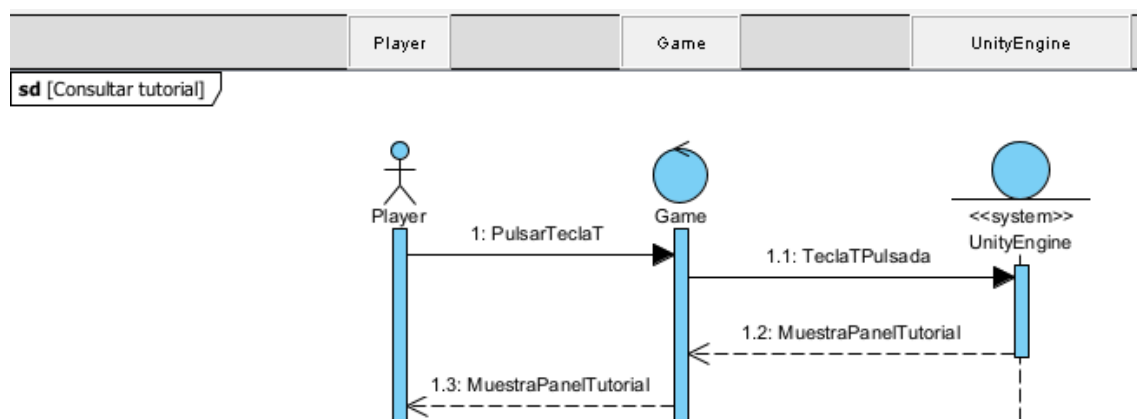


Ilustración 32: Diagrama de análisis del caso de uso "Consultar tutorial"

Durante el juego, si pulsamos T, nos mostrara el tutorial del juego.

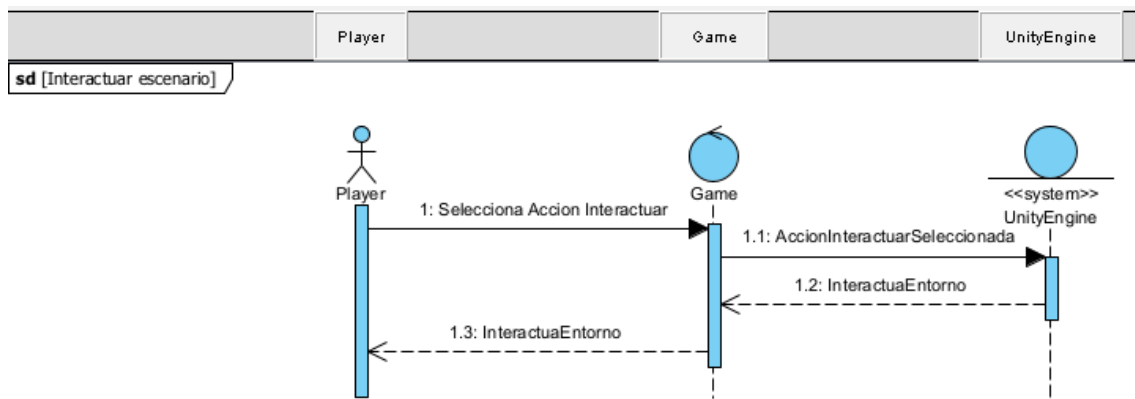


Ilustración 33: Diagrama de análisis del caso de uso "Interactuar escenario"

Cuando hemos seleccionado una unidad, podemos elegir la acción interactuar, si un objeto dentro del radio de acción de esta unidad se encuentra un objeto interactuable, gastara un punto de acción e interactuara con el objeto.

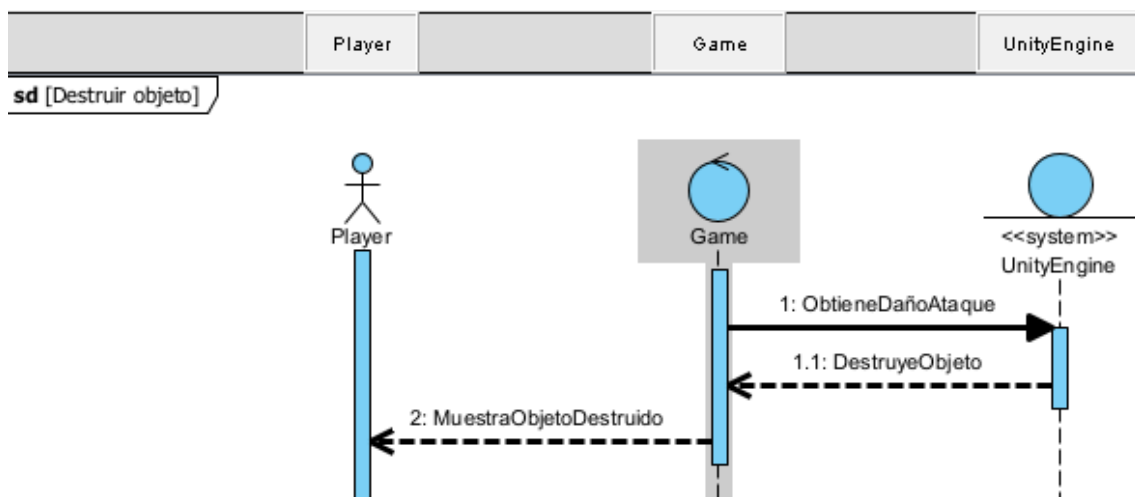


Ilustración 34: Diagrama de análisis del caso de uso "Destruir Objeto"

Por último, tenemos este caso de uso “destruir objeto” en el que si con las acciones que permiten realizar daño, provocan el suficiente en un objeto destruible, este quedara eliminado y desaparecerá de la pantalla.

6.- Plan de desarrollo, implementación y despliegue

En esta sección se presenta el plan de implementación y despliegue. Para ello, se da a conocer en primer lugar el plan de implementación, en el que se presentan las tecnologías utilizadas para la implementación del sistema, incluidas las directamente relacionadas con el funcionamiento del sistema. Posteriormente, se presenta un plan de implementación en el que se explican los entornos de ejecución necesarios para el funcionamiento del sistema.

6.1.- Plan de implementación

Para la implementación se usará el siguiente lenguaje de programación:

- C#: Lenguaje que se usa principalmente en la creación de scripts que trae por defecto Unity. Con el se ha creado la IA, las diferentes acciones que ejecutara el jugador y toda la lógica del juego.

Para el guardado de datos de la partida se usa principalmente un fichero JSON con los datos del estado del escenario en el momento de guardar para poder recuperarlo después.

6.2.- Plan de despliegue

A continuación, se va a poder apreciar la complejidad del despliegue del sistema. A parte de mostrar el diagrama, se va a describir brevemente el despliegue del proyecto:

- Usando el motor grafico de Unity para ejecutar el juego y compilarlo.
- Dependiendo del dispositivo donde se despliegue usara un entorno grafico diferente siendo en Windows donde existen dos entornos de ejecución dependiendo de la versión que tengamos instalada, aunque también se puede admitir la de Vulkan
- En MacOS se usa el entorno de ejecución propio llamado Metal.

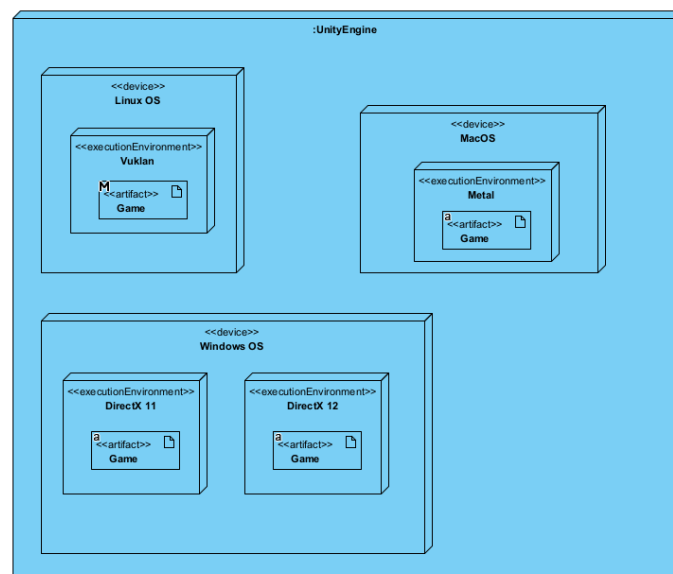


Ilustración 35: Diagrama de despliegue

7.- Conclusiones

Las conclusiones que se van a mostrar a continuación fueron tomadas después de la realización del proyecto. En general, la documentación muestra correctamente como se ha llevado a cabo la realización, pero en algunos momentos se han encontrado pequeños defectos por lo que se tuvo que modificar algún diagrama de diseño para que la implementación fuese lo más correcta posible.

Estas modificaciones han consistido en añadir clases que al principio no se encontraban expuestas en el diagrama de clases o se han modificado acciones tanto de los diagramas de casos de uso del análisis.

También comentar que la mayoría se iban refinando a medida que se iban pasando por las diferentes etapas del proyecto.