

<Super Mario Bros>

Diseño de Bajo Nivel

Versión 1.2

Producido para:

Play Store Games

Presentado por:

<Ángel Ortiz Meraz>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
28/08/23	1.0	Versión inicial del proyecto	Anngel Ortiz
14/09/23	1.1	Documentación	Anngel Ortiz
28/09/23	1.2	Documentación bajo nivel	Anngel Ortiz

1.Introducción

1.1 Propósito

El siguiente documento de Diseño de Software tiene como fin presentar la arquitectura del sistema de videojuegos Super Mario Bros a través de distintas vistas, donde se ilustran aspectos en particular del funcionamiento del software desarrollado. De esta manera, se pretende que el documento proporcione una vista detallada y comprensible, mostrando distintos diagramas UML del diseño del videojuego. Detallando además la portabilidad y estructura del proyecto.

1.2 Objetivos del Diseño

El sistema puede ejecutarse en múltiples sistemas operativos como Windows, Linux y MacOS, lo cual le otorga un alcance mayor al videojuego y a los usuarios de diversos sistemas operativos.

El videojuego cuenta con una opción de ayuda dentro del juego, en donde el jugador podrá leer las instrucciones para empezar a jugar.

El juego se controlará únicamente con el teclado. Además, jugar es muy sencillo ya que el usuario sólo necesita utilizar las teclas de flecha y la barra espaciadora para enviar bolas de fuego cuando Mario está en forma de fuego.

1.3 Definiciones, Acrónimos y Abreviaciones

Acrónimo	Definición
Acces	La relación «access» se utiliza para importaciones de paquetes privados.
Enumerator	Un enumerator (o Enum) es una clase "especial" (tanto en Java como en otros lenguajes) que limitan la creación de objetos a los especificados explícitamente en la implementación de la clase.
Interfaces	Una interfaz Java contiene una colección de métodos abstractos y

	propiedades constantes que permiten activar la herencia múltiple, es decir, que diferentes clases partan de la misma estructura.
Include	La relación <<Include>> se utiliza para incluir el comportamiento común de un caso de uso incluido en un caso de uso base para admitir la reutilización del comportamiento común.
Import	La relación «import» se utiliza para importar un paquete público.
Objetos	Un objeto es una entidad dentro de un lenguaje de programación que contiene al mismo tiempo un estado (dado por los valores que toman sus atributos) y un comportamiento (dado por una serie de métodos que permiten comunicarse con dicho objeto).
UML	Son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos).

1.4 Referencias

Documentos de software;

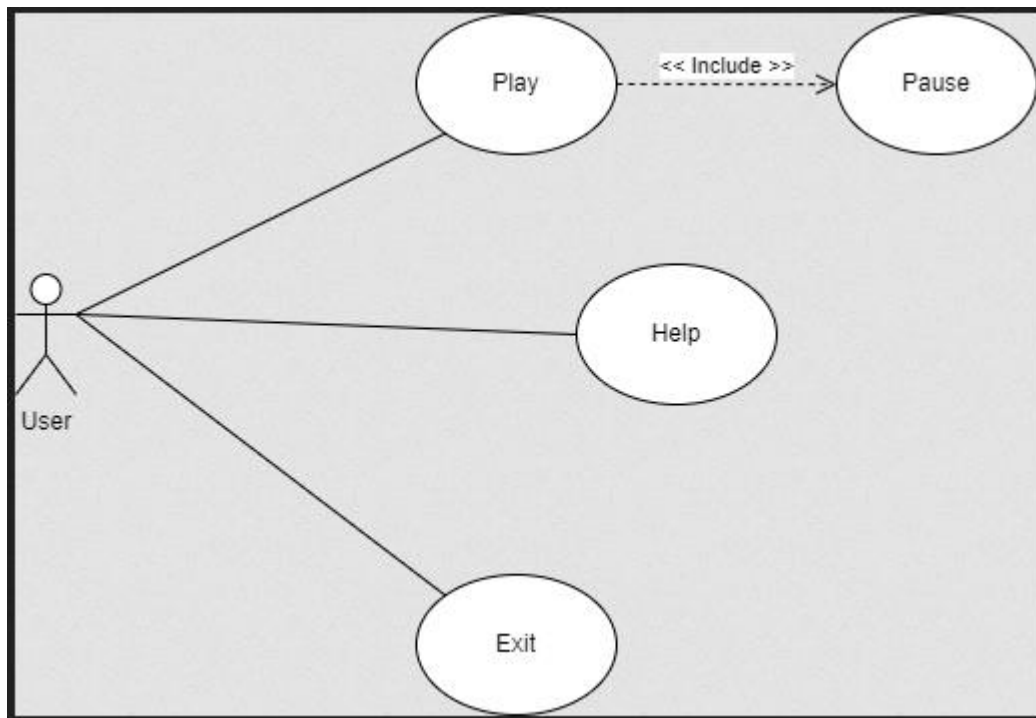
<http://www.economia.unam.mx/deschimex/cechimex/chmxExtras/repositorio/archivos/Propuesta%20Documento%20de%20Diseno%20de%20la%20DSIA.pdf>

<https://uvadoc.uva.es/bitstream/handle/10324/13198/TFG-B.724.pdf?sequence=1>

2. Diseño Detallado del Sistema

2.1 Diagrama de Casos de Uso

Un caso de uso es una técnica de modelado usada para describir lo que debería hacer un sistema nuevo o lo que hace un sistema que ya existe. Los casos de uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario, permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.



2.2 Detalle de Casos de Uso

<CU01>: <Jugar>

Actores	Usuario/Jugador	
Prioridad	Alta.	
Propósito	Empezar a jugar.	
Pre condiciones	Se debe estar en el menú de selección de mapa.	
Flujo	Paso	Acción
	1	Presionar ENTER para seleccionar un mapa a jugar.
Flujos Alternos	Paso	Acción
	1	Seleccionar START GAME en el menú principal del juego presionando ENTER.
	2	Seleccionar un mapa a jugar presionando ENTER.

<CU02>: <Pausar>

Actores	Usuario/Jugador	
Prioridad	Alta.	
Propósito	Pausar el juego.	
Pre condiciones	Se debe estar jugando (después de seleccionar mapa).	
Flujo		
	Paso	Acción
	1	Presionar ESC para pausar el juego.
Flujos Alternos	Paso	Acción
	1	Seleccionar START GAME en el menú principal del juego presionando ENTER.
	2	Seleccionar un mapa a jugar presionando ENTER.
	3	Empezar a jugar ya seleccionado el mapa.

<CU03>: <Ayuda>

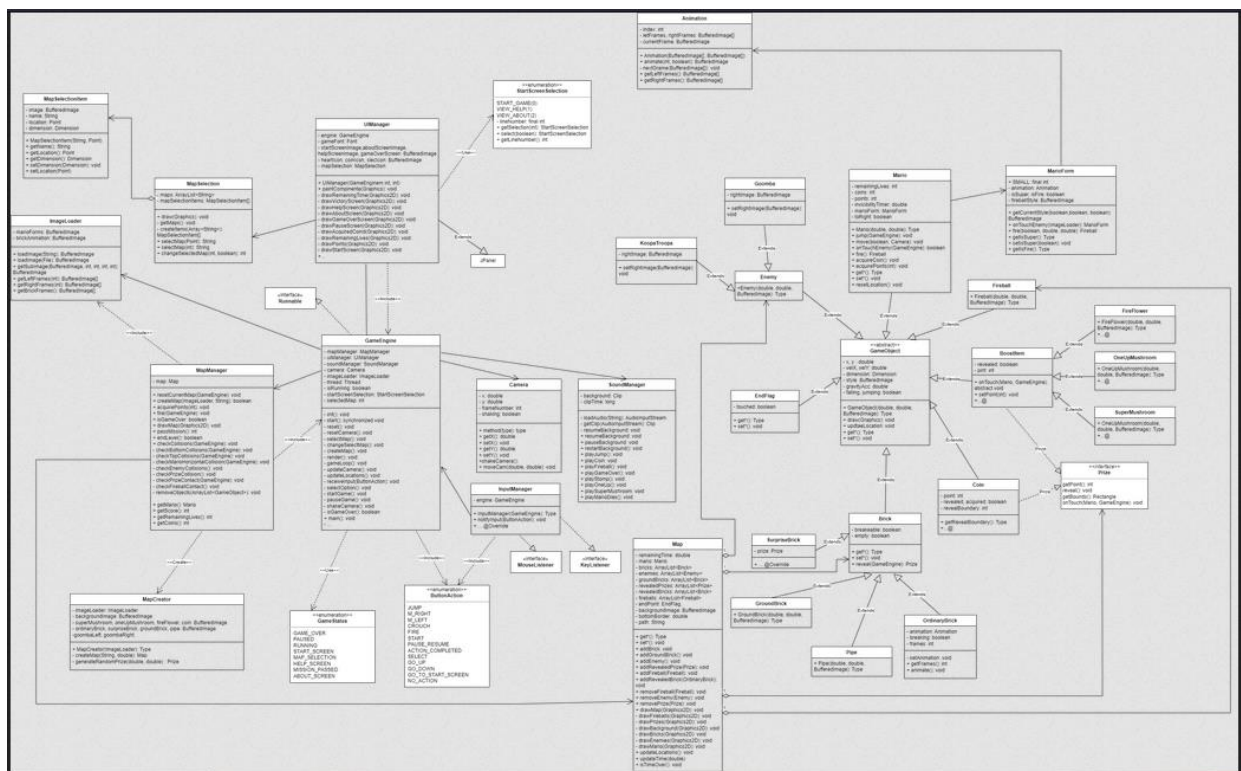
Actores	Usuario/Jugador	
Prioridad	Media.	
Propósito	Explicar al jugador los controles de juego.	
Pre condiciones	Se debe seleccionar HELP en el menú principal de juego.	
Flujo		
	Paso	Acción
	1	Seleccionar HELP con ENTER en el menú principal del juego.
Flujos Alternos		

<CU04>: <Salir>

Actores	Usuario/Jugador	
Prioridad	Alta.	
Propósito	Salir del juego.	
Pre condiciones	Se debe estar jugando (después de seleccionar mapa).	
Flujo		
	Paso	Acción
	1	Presionar X roja para salir el juego.
Flujos Alternos		

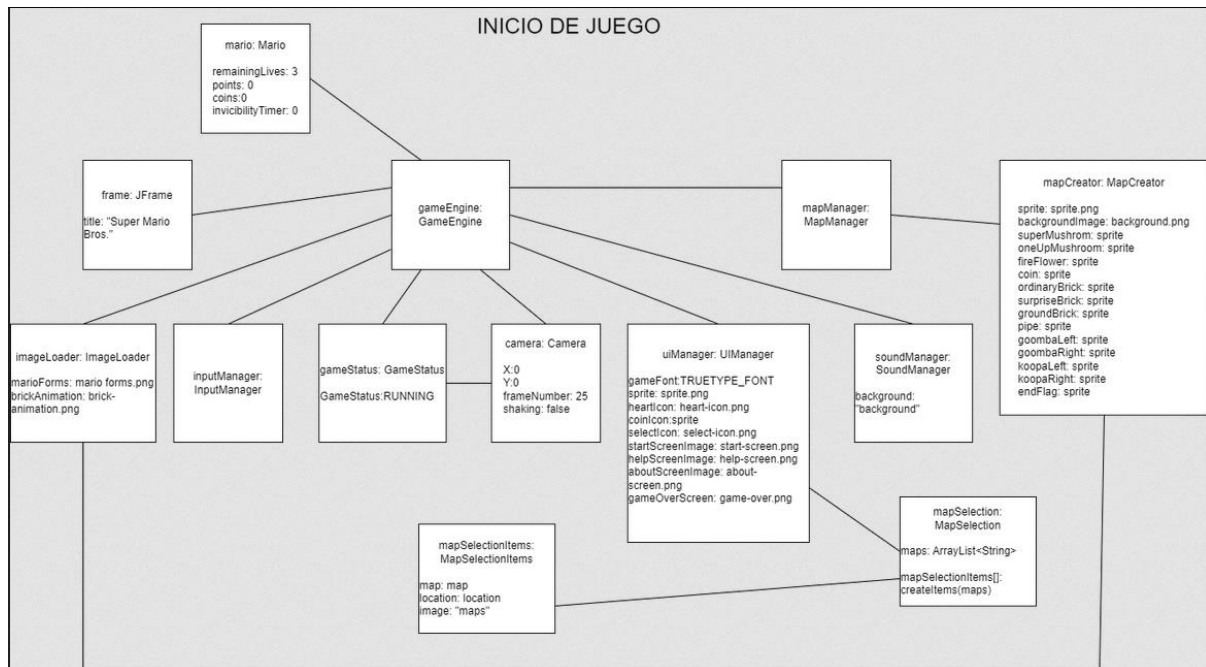
2.3 Diagrama de Clases

Se presentan las clases que componen al sistema, así como las interfaces, enumerador, además de especificar cómo se relacionan entre ellas para cumplir el objetivo del videojuego. De esta manera se presenta una vista del comportamiento de la lógica del sistema y la creación de objetos, así como sus propiedades y relaciones entre ellas.

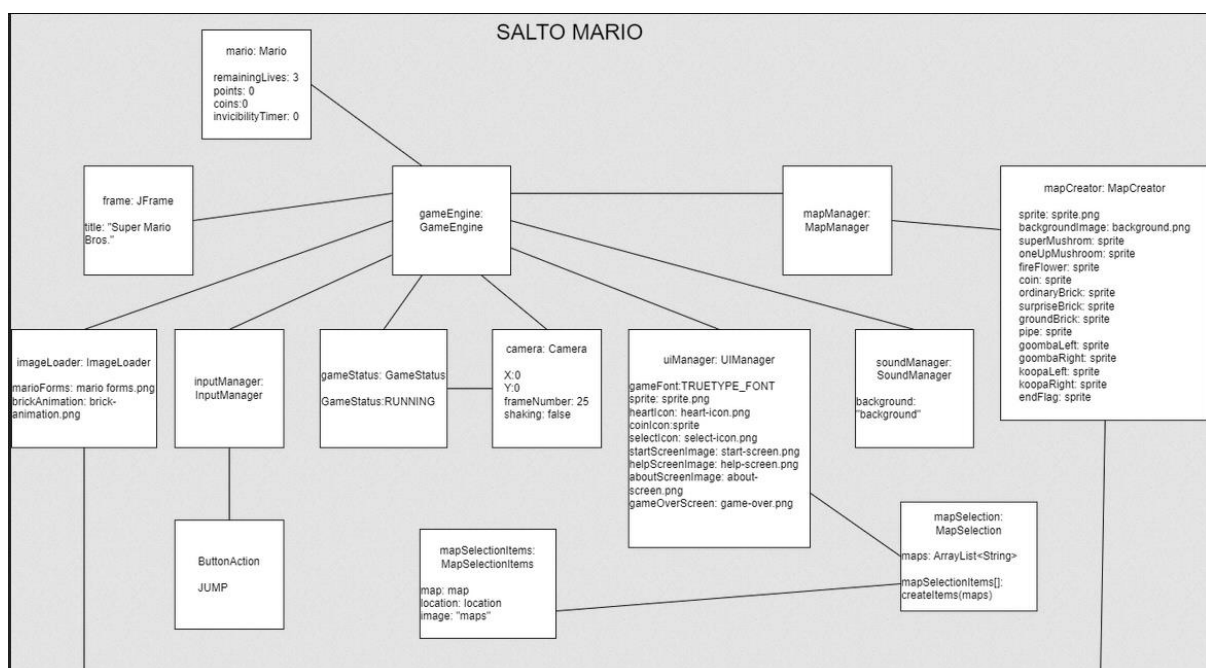


2.4 Diagrama de Objetos

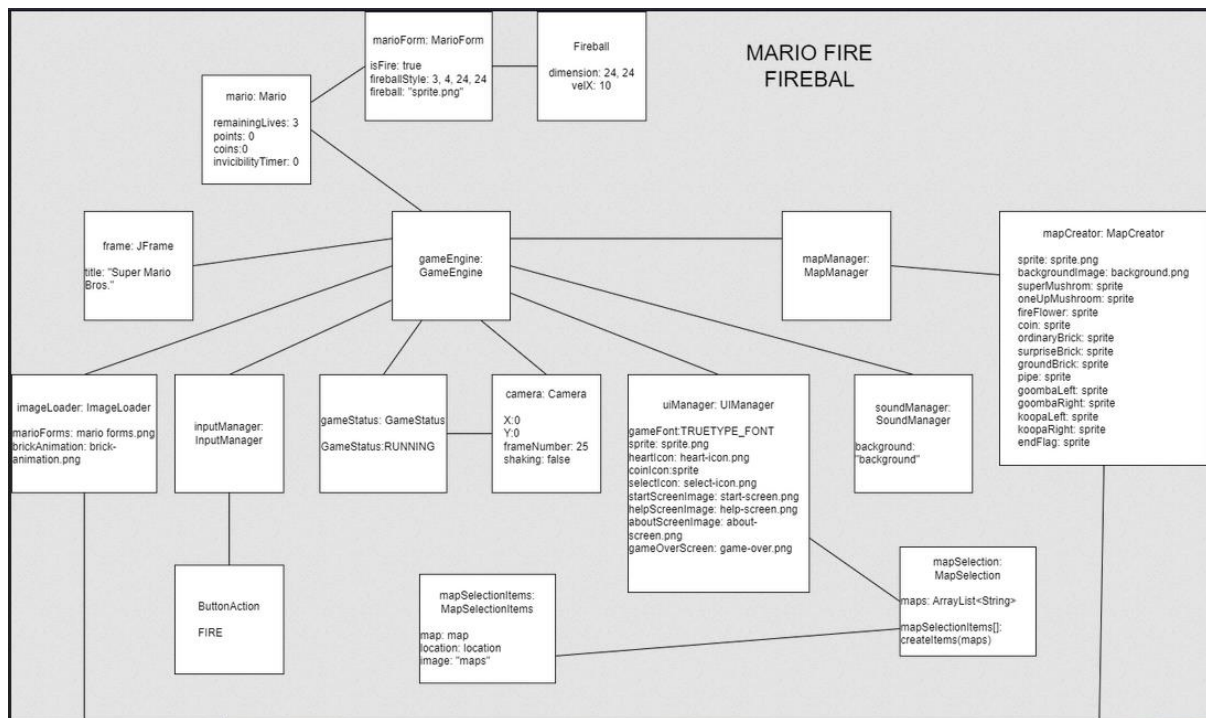
Un diagrama de objetos UML representa una instancia específica de un diagrama de clases en un momento determinado en el tiempo. Cuando se lo representa visualmente, verás muchas similitudes con el diagrama de clases. Un diagrama de objetos se enfoca en los atributos de un conjunto de objetos y cómo esos objetos se relacionan entre sí.



Se presenta un diagrama de objetos detallándose en el momento en que el videojuego genera o crea el mapa, los personajes y los elementos dentro de él, se muestran cuáles son los objetos que se encuentran instanciados en este momento y cuáles se están relacionando.



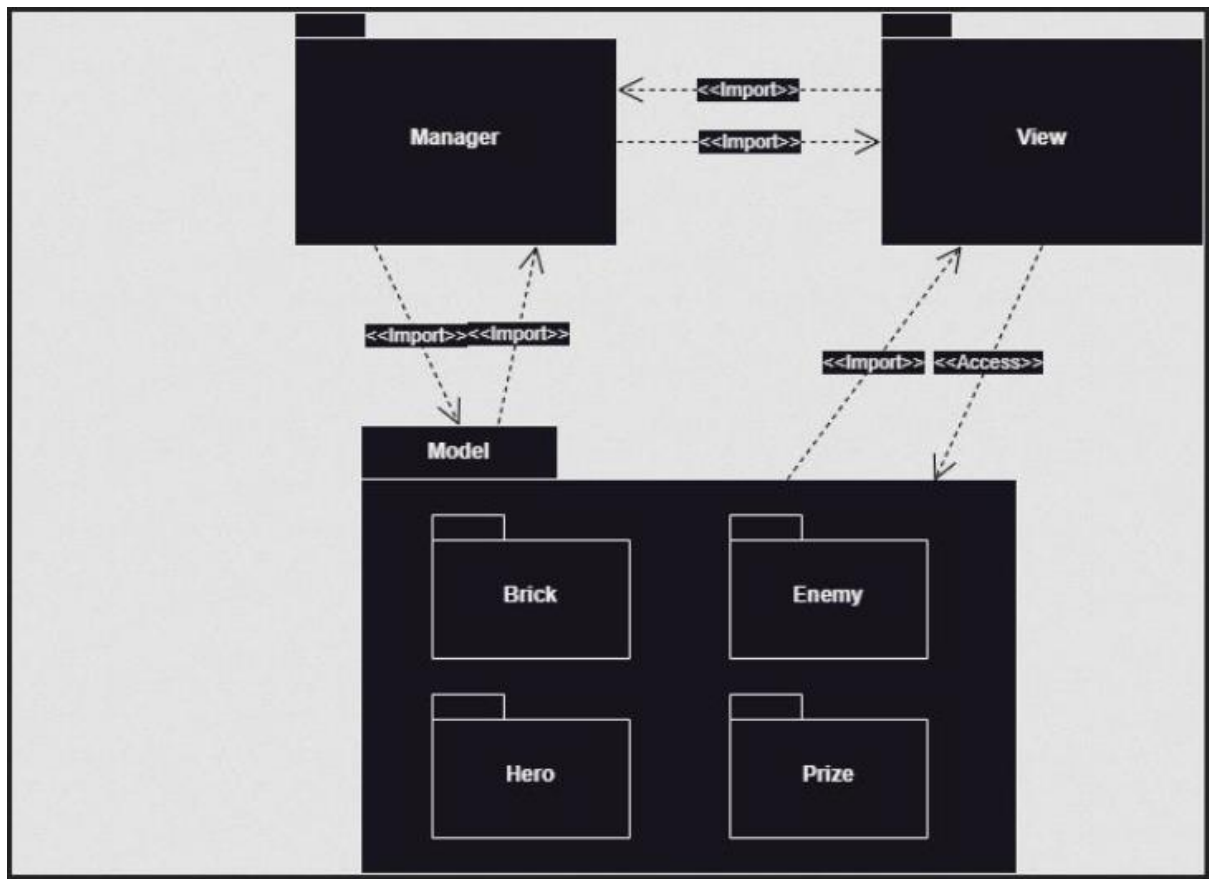
Se presenta un diagrama de objetos detallándose en el momento en que el jugador ya se encuentra manipulando al personaje principal del juego y éste genera un salto, se muestran cuáles son los objetos que se encuentran instanciados en este momento y cuáles se están relacionando.



Se presenta un diagrama de objetos detallándose en el momento en que el jugador ya se encuentra manipulando al personaje principal del juego y éste adquiere una nueva forma de Mario, el cual ahora lanza bolas de fuego, se muestran cuáles son los objetos que se encuentran instanciados en este momento y cuáles se están relacionando.

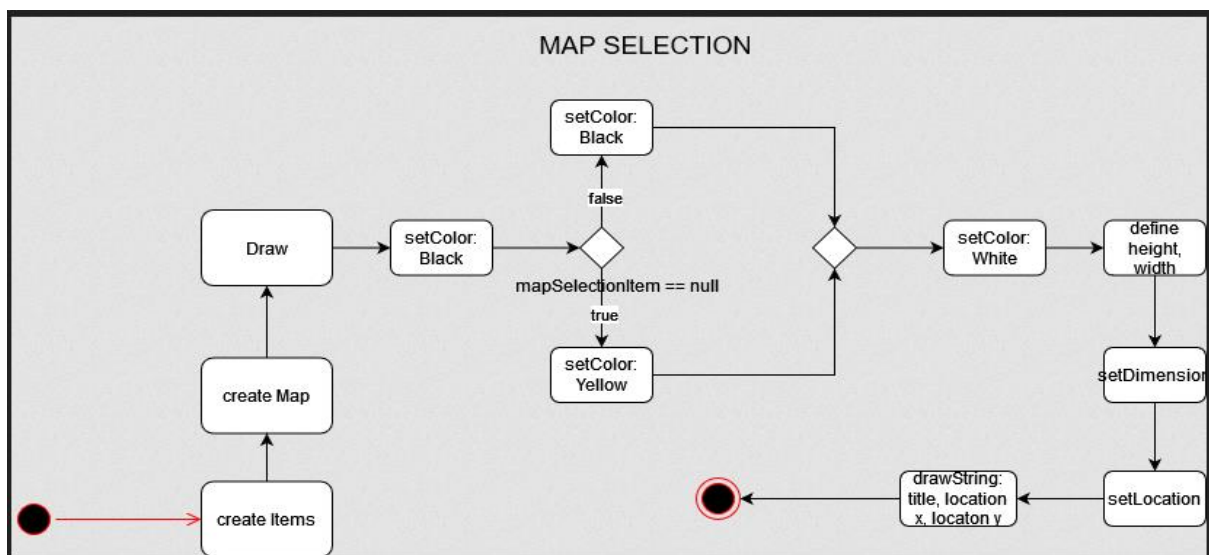
2.5 Diagrama de Paquetes

Se presentan los paquetes que conforman el sistema y cómo interactúan entre sí, en el videojuego tenemos tres paquetes principales que se encargan de controlar y visualizar los modelos del juego, donde éste último paquete contiene otros cuatro paquetes dentro, los cuales almacenan la información de algunos objetos dentro del juego, personajes y enemigos.

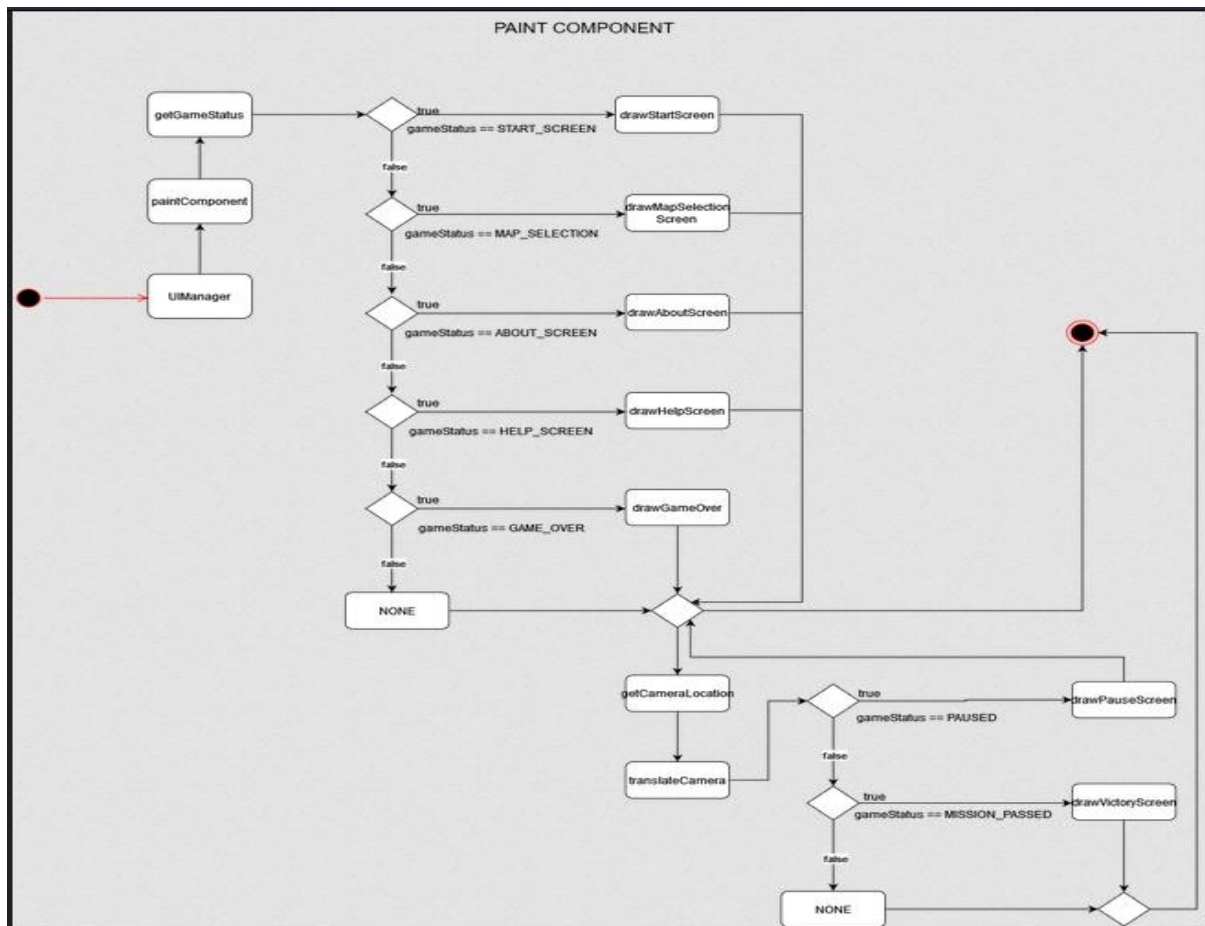


2.6 Diagrama de Actividad

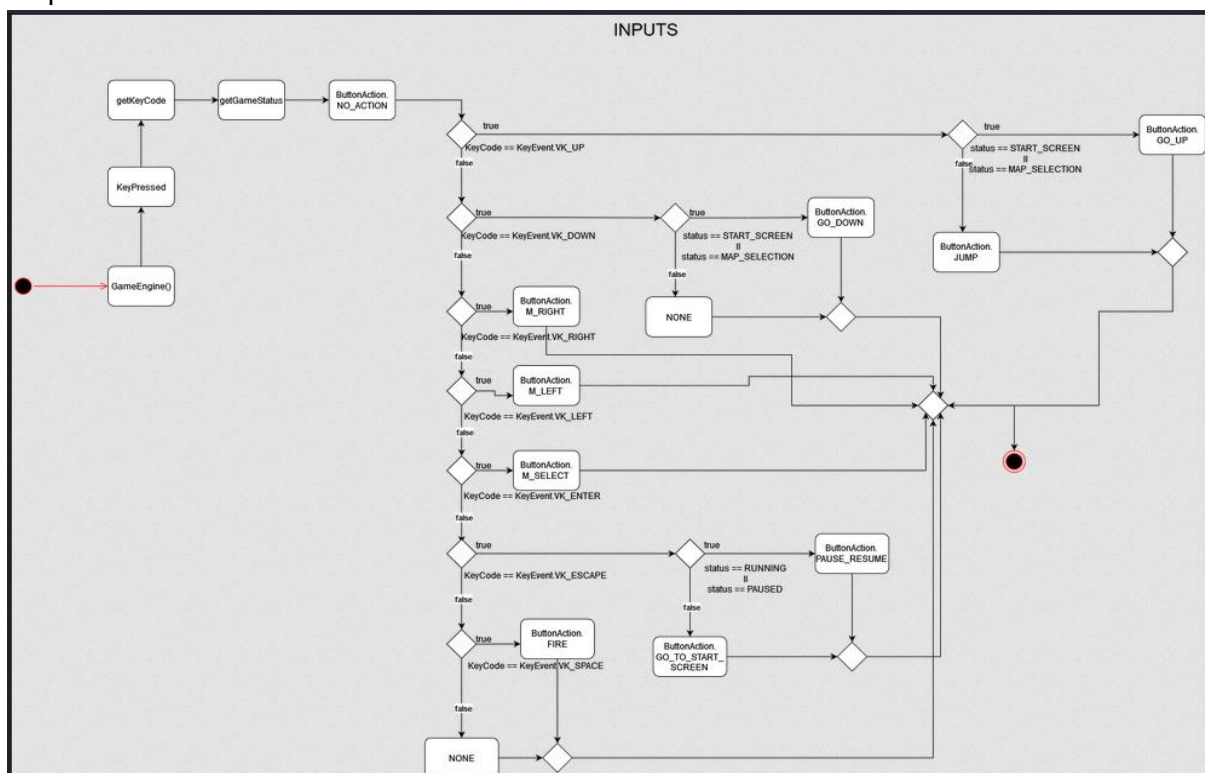
En UML un diagrama de actividades se usa para mostrar la secuencia de actividades. Los diagramas de actividades muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Estos también pueden usarse para detallar situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas actividades.



Se muestra cuál es la secuencia de actividad para seleccionar un mapa.



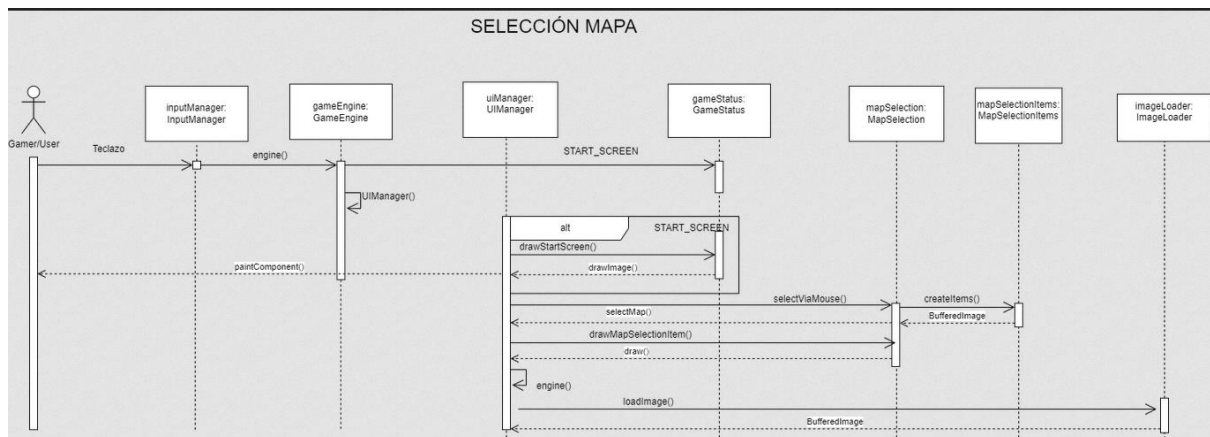
Se muestra cuál es la secuencia de actividad para dibujar los componentes al crear un mapa.



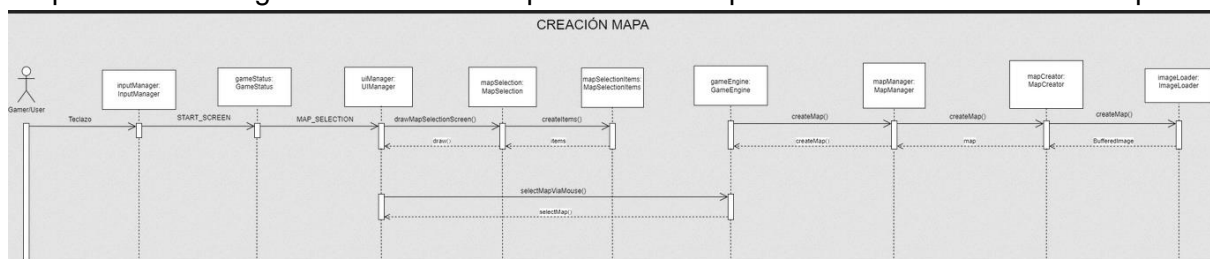
Se muestra cuál es la secuencia de actividad para los diferentes inputs que se recibirán del usuario.

2.7 Diagrama de Secuencia

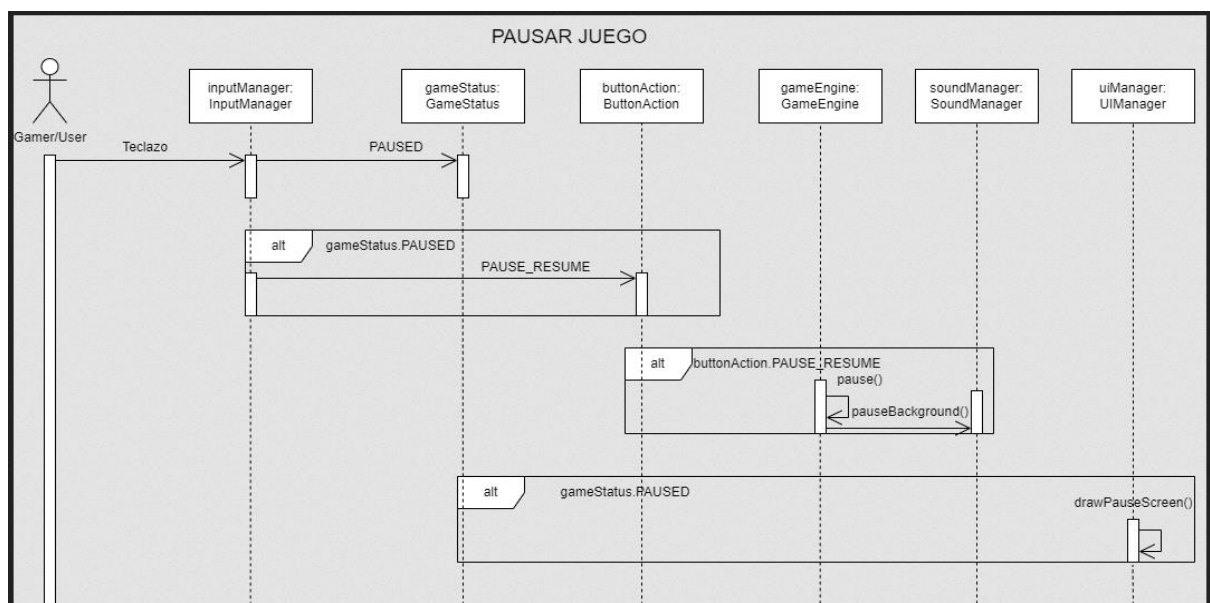
Un diagrama de secuencia es un tipo de diagrama de interacción porque describe cómo y en qué orden un grupo de objetos funcionan en conjunto. Tanto los desarrolladores de software como los profesionales de negocios usan estos diagramas para comprender los requisitos de un sistema nuevo o documentar un proceso existente.



Se presenta un diagrama de secuencia para detallar el proceso de selección de un mapa.



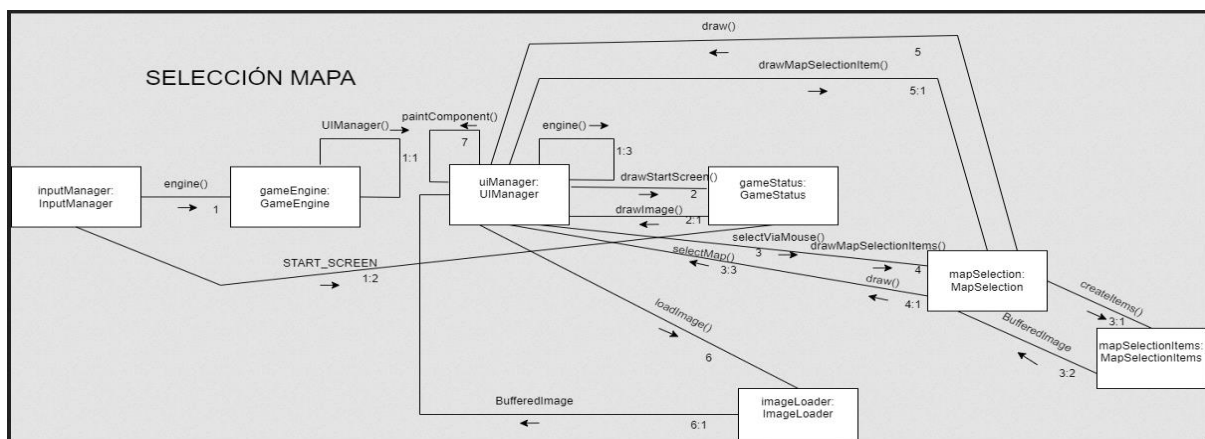
Se presenta un diagrama de secuencia para detallar el proceso de selección de un mapa.



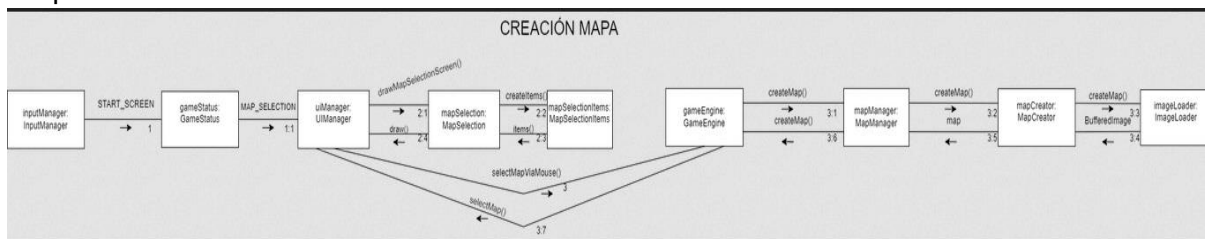
Se presenta un diagrama de secuencia para detallar el proceso de pausa del juego.

2.8 Diagrama de Comunicación

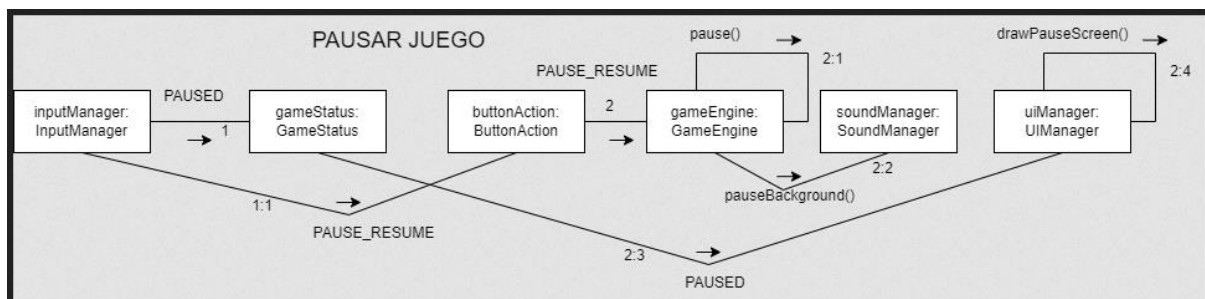
Un diagrama de comunicaciones, inicialmente llamado un diagrama de colaboración, es un diagrama de interacción que muestra información similar a los diagramas de secuencia, pero su foco principal es en la relación de objetos. En los diagramas de comunicaciones, los objetos como se muestran con conectores de asociación entre ellos. Los mensajes se agregan a las asociaciones y se muestran como flechas cortas apuntando en la dirección del flujo del mensaje. La secuencia de los mensajes se muestra a través de un esquema enumerado.



Se presenta un diagrama de secuencia detallando el proceso de selección de mapa, mostrando la interacción entre los diversos objetos involucrados en el proceso y enumerando los pasos en secuencia.



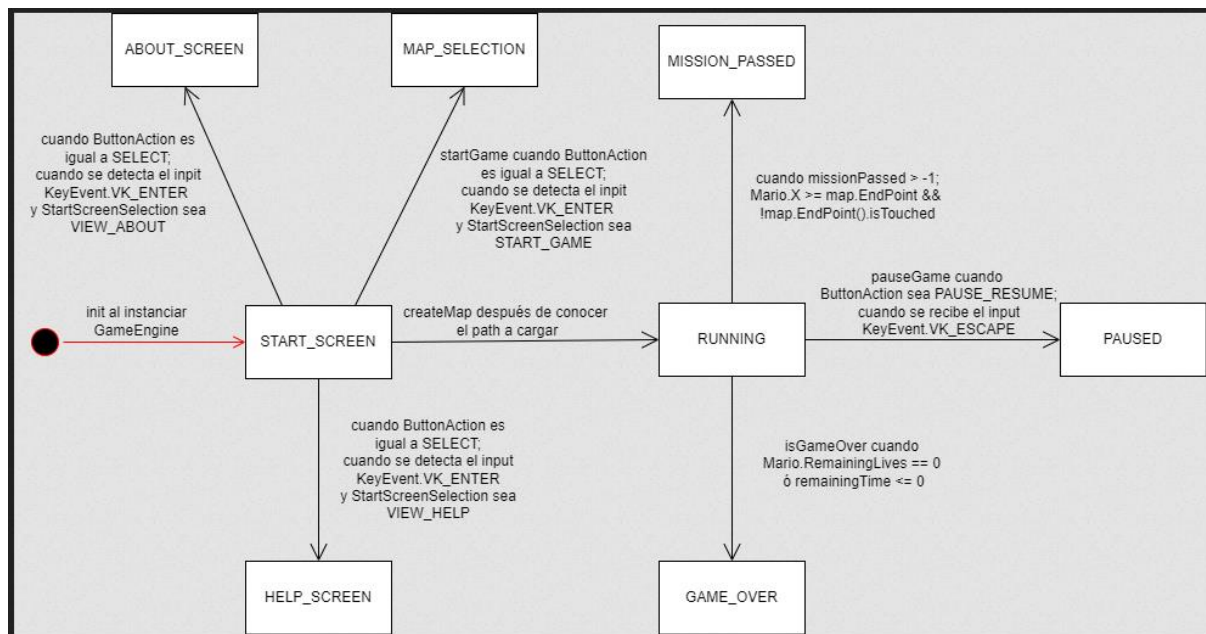
Se presenta un diagrama de secuencia detallando el proceso de pausa del juego, mostrando la interacción entre los diversos objetos involucrados en el proceso y enumerando los pasos en secuencia.



Se presenta un diagrama de secuencia detallando el proceso de pausa del juego, mostrando la interacción entre los diversos objetos involucrados en el proceso y enumerando los pasos en secuencia.

2.9 Diagrama de Máquina de Estados

Una máquina de estados es cualquier dispositivo que almacena el estado de un objeto en un momento dado y puede cambiar el estado o causar otras acciones según la entrada que reciba. Estados se refiere a las diferentes combinaciones de información que un objeto puede mantener, no la forma en que el objeto se comporta.



Se presenta un diagrama de estados sobre el enumerador Game Status, donde se detalla los distintos estados que puede adoptar el estatus del videojuego, así como cuáles son las condiciones y en qué momento pasa de un estado a otro durante la ejecución del sistema.