

Entornos Inmersivos, Interactivos y de Entretenimiento

Grado en Ingeniería Informática

Grupo 2C - ISS Manticore

Autor:

Elena Goyanes González

Rocío Rego Sierra

Martiño Moure Vila

Welton Vieira dos Santos

Juan Miguel Regal Llamas

Tutor/es:

Julián Alfonso Dorado De La Calle (julian.dorado@udc.es)

Enrique Fernández Blanco (enrique.fernandez@udc.es)

f

facul
infor
da c

Entornos Inmersivos, Interactivos y de Entretenimiento

Memoria Artística

Autor

Elena Goyanes González
Rocío Rego Sierra
Martíño Moure Vila
Welton Vieira dos Santos
Juan Miguel Regal Llamas

Tutor/es

Julián Alfonso Dorado De La Calle (julian.dorado@udc.es)
Ciencias de la Computación y Tecnologías de la Información
Enrique Fernández Blanco (enrique.fernandez@udc.es)
Ciencias de la Computación y Tecnologías de la Información



facultade de
informática
da coruña

Grado en Ingeniería Informática



A Coruña, 21 de Abril de 2021

Índice general

Índice de figuras

1 Desarrollo Artístico

1.1 Antecedentes

En el año 2500 la Tierra era un yermo vacío y la humanidad la abandonó, mandando robots autoconscientes steampunk a explorar el espacio, ahora vivían colonizando el espacio. Figura ??.



Figura 1.1: Tierra devastada

1.2 Ambientación

La historia se transcurre en una enorme y antigua nave en la que suceden cosas extrañas, de la que nuestro protagonista tendrá que huir. ISS Manticore (Figura ??) es la nave misteriosa donde nuestro protagonista tenía que entregar un paquete.

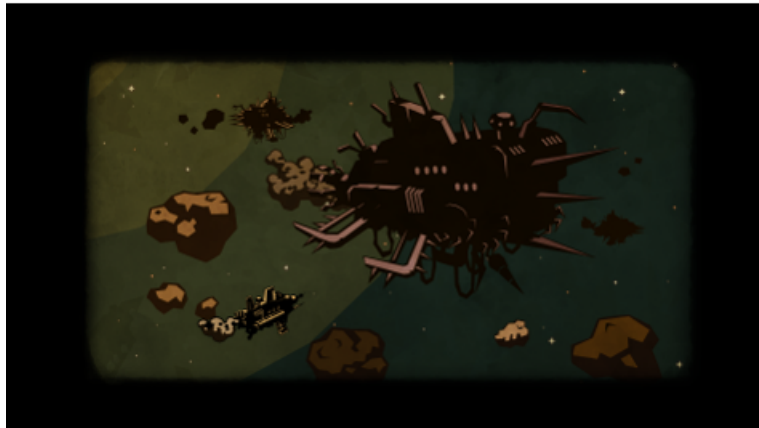


Figura 1.2: ISS Manticore

1.3 Historia

Jackie (Figura ??) es un cartero espacial que tiene que entregar un paquete en una misteriosa nave, pero cuando llega no hay nadie, y acaba encerrado en la misteriosa nave de la que tendrá que escapar. Cuando nuestro protagonista abre el paquete, verá que es un arma de fuego, la tendrá que usar para abrirse paso a través de la nave contra los misteriosos moradores que se vaya encontrando en ella.

1.4 Personajes

- **Jackie:** Nuestro protagonista, no tiene muchas luces, pero sabe sobrevivir



Figura 1.3: Aspecto de Jackie

- **Enemigos:** En esta enorme nave habitan unos seres que al parecer no les gusta mucho la compañía de otros, por lo que te atacaran si te cruzas en su camino.
- **Octopus:** Son unos seres extraños fruto de un experimento de crear inteligencia artificial utilizando cerebros biológicos.



Figura 1.4: Aspecto de un Octopus

- **Tortuga:** Alguien se olvidó de alimentar a las mascotas de la nave, pero por alguna razón se siguen moviendo, y parecen hambrientas.



Figura 1.5: Aspecto de una Tortuga

- **Spiderbots:** Alguien que se aburría en esta nave se le ocurrió la brillante idea de ponerle patas a las torretas, y parece que muy bien no salió.



Figura 1.6: Aspecto de un Spiderbot

1.5 Jugabilidad

Side scroller en el que completas niveles estilo metroidvania, en el que el personaje pelea:

- Utilizando un arma de fuego (escopeta - Figura ??), en el que avanzamos de forma similar a juegos como “My Friend Pedro” avanzando por diferentes niveles acabando con los enemigos(Figura 1.6), con una estética de enemigos y de ambientación ligeramente steampunk.



Figura 1.7: Ejemplo de la escopeta



Figura 1.8: Ejemplo de un posible escenario y la dinámica de disparos en tempo real

1.6 Items para el personaje

- **Vidas:** Para recuperar vida y ayudar al jugador e incentivarlo a explorar el escenario. (Figura ??)



Figura 1.9: Imagen de item para recuperar a vida

- **Wrench:** Items coleccionables necesarios para completar el segundo nivel en el que tienes que recoger piezas para reparar el motor de la nave y así también hacer al jugador explorar el escenario (Figura ??).



Figura 1.10: Imagen de wrench representada por una llave inglesa girando

1.7 Lugares

1.7.1 ISS Manticore

La nave en la que se desarrolla el juego, está llena de misterios y seres extraños poco amistosos tiene diferentes zonas y salas dentro, en este juego se incluyen 3 salas, que corresponderían a 3 niveles.

1.7.2 Sala de calderas de la nave

El primer lugar al que llega el protagonista, en esta zona de la nave se encuentran todo el sistema del motor de la nave.

1.7.3 Bodega de carga

Es la bodega de carga de la nave, donde está el almacén y tiene un montón de extraños trastos viejos de la nave y muchas herramientas que serán de utilidad al protagonista.

1.7.4 Puente de mando

La última sala que el protagonista deberá atravesar, y así conseguir al fin huir de la nave.

1.8 Guion

1.8.1 Fase 1 - Sala de calderas

La escena empieza con el cartero intentando entregar ese supuesto paquete a la nave y al llegar cerca de la nave el personaje **Jackie** se da cuenta que en esa nave no hay nadie que pueda responder y él toma la decisión de entrar en la nave y él mismo se queda atrapado.

Para salir de esa nave el tiene que buscar otra salida. Jackie decide abrir el paquete y se dentro del mismo se encuentra un arma y decide seguir hacia delante para intentar localizar otra salida.

1.8.2 Fase 2 - Almacen

Esa escena empezará con unos de los enemigos dañando un motor de la sala de calderas lo que obliga a nuestro personaje (Jackie) a intentar localizar piezas para reparar ese motor dañado avanzando a este siguen escenario que será el almacen de la bodega de carga. Nuestro personaje encuentra herramientas y piezas al recorrer el camino que pueden ser utilizadas para reparar el motor. Además de ir derrotando los enemigos que van apareciendo a medida que Jackie avanza. Para terminar la fase, Jackie tiene que haber recogido todas las piezas que se encuentra en esa fase para reparar el motor.

1.8.3 Fase 3 - Casco superior de nave

Esa fase se empieza una batalla final con los últimos enemigos que protegen la mesa de mando donde nuestro personaje tiene que enchufar una tarjeta, que contiene un chip que permite Jackie efectuar un pedido de auxilio. Al momento que Jackie enchufa la tarjeta, la nave sufre un apagón debido a daños causados anteriormente por los enemigos.

1.9 Videojuego en 3D

1.9.1 Fases

Fase 1 - Motores exteriores de la nave

Nuestro protagonista, tiene que salir a arreglar el motor de la nave para recuperar la electricidad y que se envíe la llamada de ayuda, así que tendrá que salir fuera de la

nave (por lo que no hay gravedad) , y con su pistola gancho tendrá que ir avanzando por la turbinas girando y demás elementos flotando hasta llegar al motor a arreglar

Fase 2 - Zonas de motores

El protagonista se tiene que defender de enemigos mientras repara el motor.

Fase 3 - Espacio exterior

Una nave para salvarle aparece, nuestro protagonista tendrá que ir escalando entre basura espacial y cajas (de la sala de carga) flotando en el espacio entre las dos naves para poder huir y alcanzar a la otra nave. (las torretas de la nave de la que huye le pueden ir disparando mientras para que no huya).

2 Desarrollo técnico

2.1 Videojuego en 2D

2.1.1 Descripción

ISS Manticore es un scroll lateral con ligeros tintes de metroidvania¹ en el diseño de escenarios al tener algunos incentivos para desviarse del camino principal, pero dividido en niveles donde los jugadores comienzan cada fase en el extremo izquierdo de un escenario lineal.

Su objetivo es alcanzar la salida de la nave de la cual se ha quedado atrapado en el momento que ha intentado entregar un paquete en la misma.

El protagonista va avanzando y derrotando los enemigos que contiene en cada fase.

2.1.2 Personajes

Jackie, el protagonista del juego tendrá que completar distintas fases, en las que tendrá que superar distintos enemigos y acometer distintas tareas para alcanzar su objetivo, para ello el jugador tendrá que mover al protagonista empleando las flechas del teclado para avanzar, esquivar a los enemigos y recoger los coleccionables necesarios y disparando con un click de ratón a los distintos personajes que se interpongan en su camino.

¹Metroidvania: es un subgénero de juego basado en un concepto de plataformas no lineal con un mundo conectado que fomenta que el jugador lo explore

2.1.3 Enemigos

Los enemigos se interponen en el camino del protagonista habitando los diferentes escenarios por los que avanza dañando al protagonista si este entra en contacto con los enemigos.

- **Octopus:** Estos pequeños seres extraños se encuentran levitando en una misma posición con el objetivo de interrumpir al protagonista en su camino para completar su cometido. Ejemplo en la Figura ??.
- **Tortugas:** Están en constante movimiento de un lado a otro y dificultan el transcurso del personaje, que debe abatirlos o esquivarlos, para no salir dañado y conseguir completar su misión. Ejemplo en la Figura ??.
- **Spiderbots:** Son el enemigo más difícil de derrotar para el protagonista debido a que dispara proyectiles cada 10 segundos que pueden dañarlo si no es capaz de esquivarlos. Ejemplo en la Figura ??.

2.1.4 Diseño

Patrón estrategia

Este patrón ha sido utilizado en la gran mayoría del código, con la idea de abstraer en clases una serie de estados y comportamientos y, a medida que se se van extendiendo otras clases de una forma bastante jerárquica, haciendo que el comportamiento sea cada vez más específico, por ejemplo el Menu principal del juego que está compuesto por varios componentes como imágenes, botones (estilo rolover) para seleccionar las opciones pertinentes.

Patrón Singleton

El patrón singleton está presente en la clase Director, GestorRecursos y tambien en las factorías de los personajes y de las fases del juego. Además de solamente permitir

que se instancia solamente un objeto con el uso de clases internas en el caso del director, que solamente tiene uno en todo el proceso.

Personajes

La implementación de los personajes hereda de la clase MiSprite y va jerarquizando como se muestra en la Figura ?? . Esa clase incorpora los elementos necesarios para almacenar las posiciones, sprites, velocidades y todos los elementos comunes como el scroll.

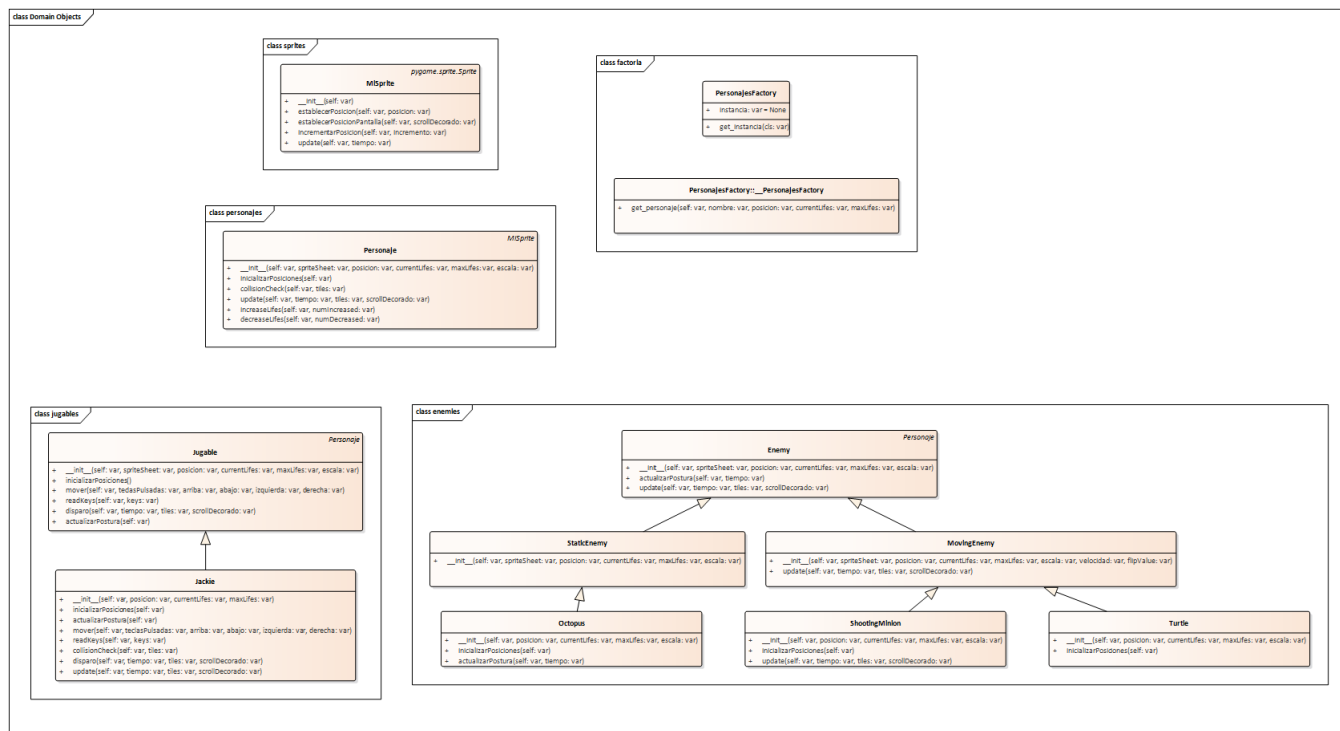


Figura 2.1: Ejemplo de la jerarquía de la clase Personaje

Cada objeto del tipo personaje es instanciado por su factoría correspondiente con la intención de permitir un mejor mantenimiento y expansión del código futuramente.

Esos personajes están diferenciados por dos tipos básicos, uno jugable y otro los enemigos.

Fases

La implementación de la dinámica de fases es muy similar a la dinámica de los personajes como se puede apreciar en la Figura ?? conservando los métodos “update”, “eventos” y “dibujar” de la clase Escena.

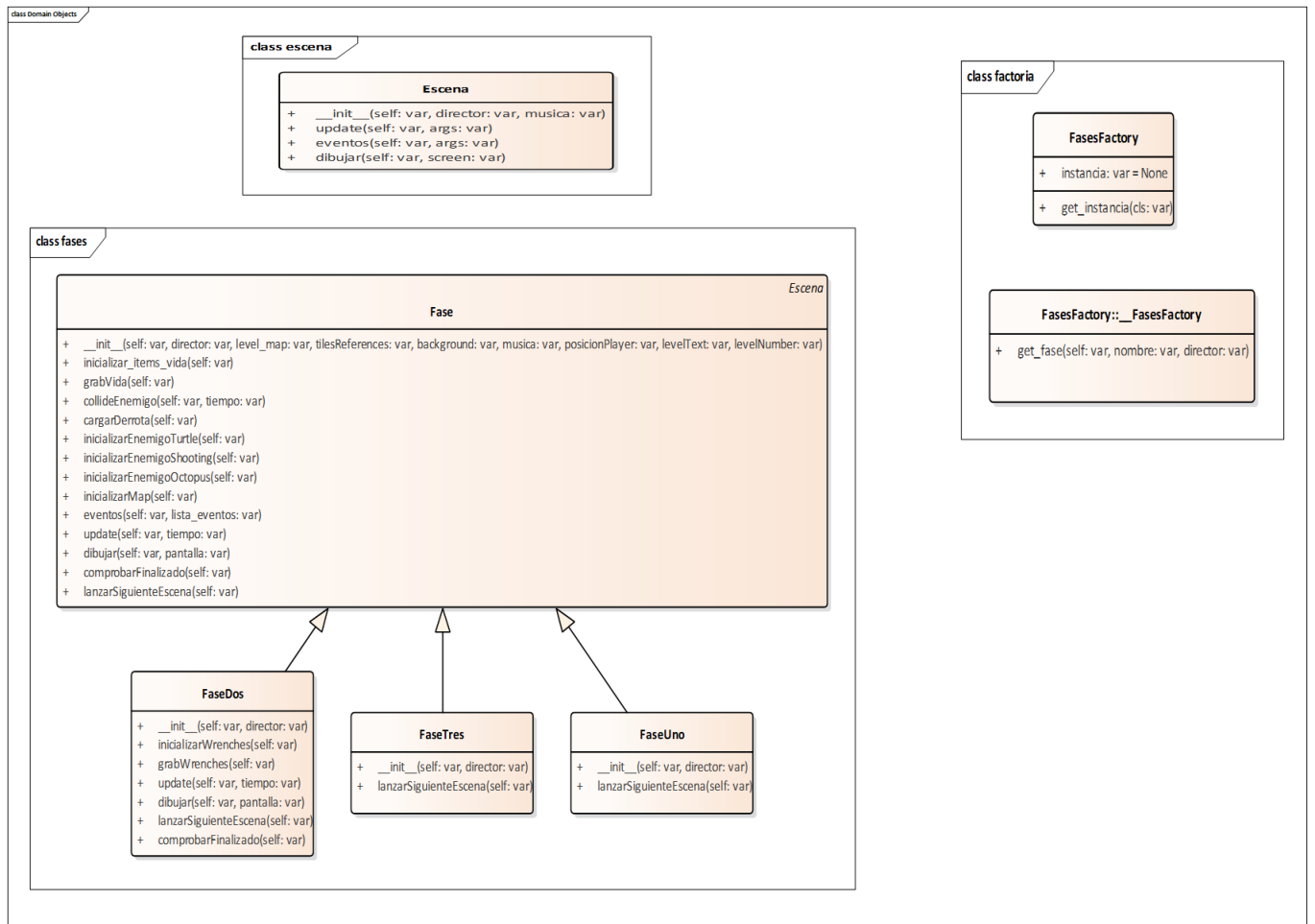


Figura 2.2: Ejemplo de la jerarquía de la clase Fase

Escena

La clase Escena define toda la estructura visual del juego, donde desde ahí se puede controlar los menus, personajes, transiciones y etc.

La clase Escena posee tres métodos principales:

- eventos: Encargado de leer los eventos producidos por la interacción del usuario con el sistema.
- update: Encargado de actualizar el modelo de escena en cuestión basado en los eventos producidos durante la interacción del usuario.
- dibujar: Encargado de dibujar los elementos de la parte visual del juego.

Director

El director es el encargado de ejecutar la escena pertinente, que puede ser un menú o las fases correspondientes de las determinadas etapas del juego. La Figura ?? muestra su estructura.

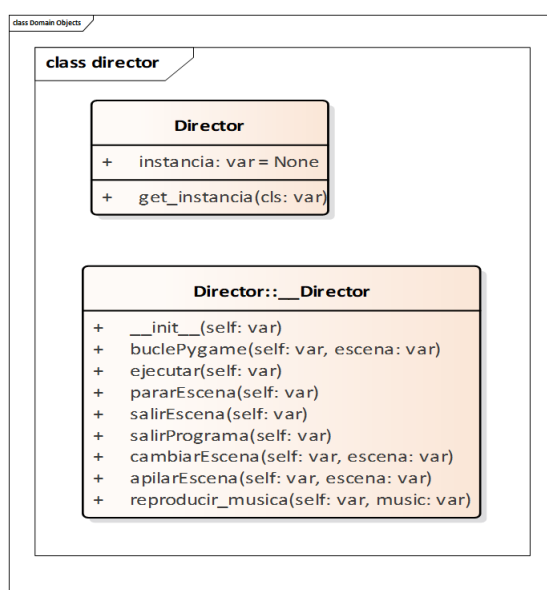


Figura 2.3: Ejemplo de la jerarquía de la clase Director

Como se puede observar, director posee una clase interna.

Gestor de recursos

Como su propio nombre dice, es el encargado de suministrar los recursos necesarios para la una buena interacción con el juego. La Figura ??

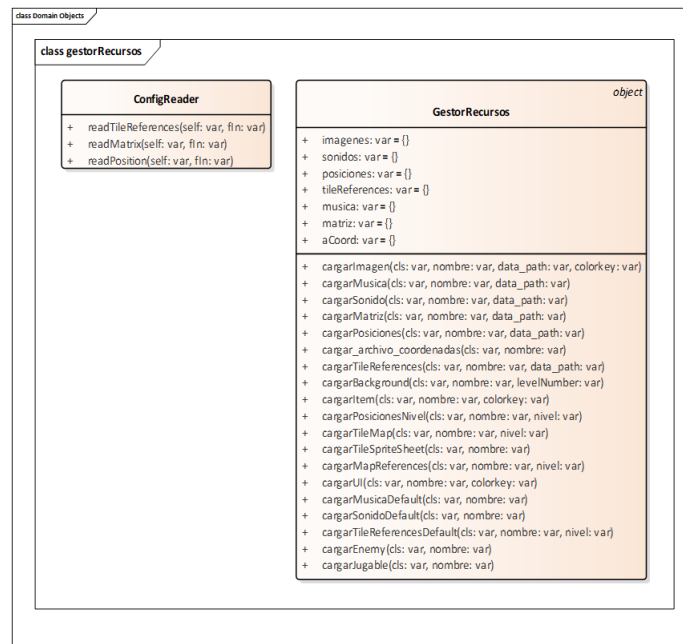


Figura 2.4: Ejemplo uml del gerenciador de recursos

2.1.5 Escenas

La transición entre escenas se muestra en el diagrama de la Figura ??.

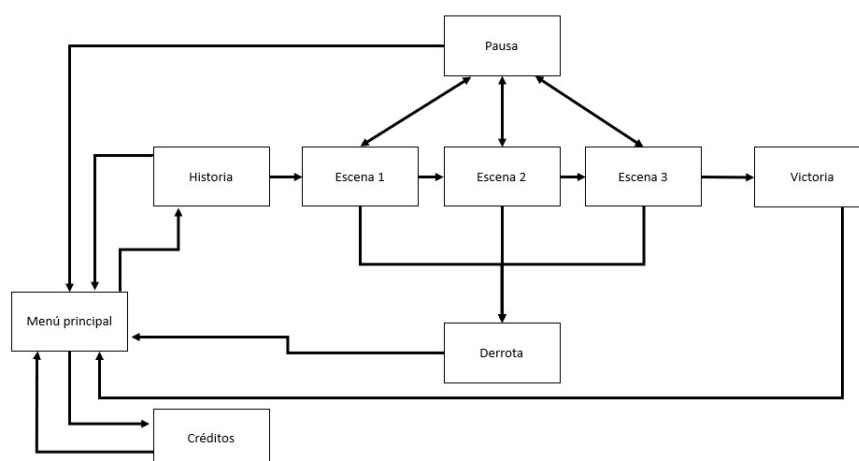


Figura 2.5: Diagrama de transición de las escenas

Menú principal

En esta primera pantalla (Figura ??), que se muestra al jugador nada más ejecutar el juego, se le permite elegir entre comenzar con la historia, visualizar los créditos, lo que enviaría al usuario a la pantalla de Créditos, o la opción de salir, que cerraría el juego.

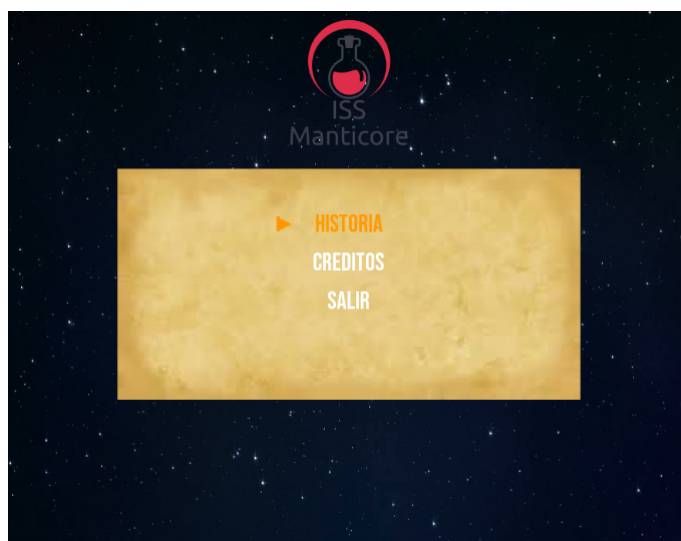


Figura 2.6: Ejemplo del menu principal

Historia

Esta pantalla (Figura ??) muestra un breve resumen de la historia y la leyenda del juego, en la que se puede ver la configuración inicial que contiene la imagen del personaje, el número de vidas inicial, el número de proyectiles con el que cuenta antes de comenzar la aventura (con miras a añadir más adelante objetos de munición y diferentes armas o proyectiles). Desde esta pantalla se puede volver al Menú principal pulsando la tecla Esc o pasar a la Escena 1 pulsando la tecla Enter.



Figura 2.7: Ejemplo del apartado historia del menu principal

Fase 1 - Sala de calderas

La escena comienza con el protagonista en la sala de calderas, este emprende su aventura con tres vidas, que podrá aumentar o disminuir segundo se desenvuelva, nada más arrancar aparece el primer enemigo, un spiderbot al que tendremos que derrotar, si seguimos avanzando nos encontramos con el segundo enemigo, en este caso un octopus, y también con el primer botiquín, que nos permitirá ganar una vida, continuando con la aventura nos encontraremos con más enemigos a los que tendremos que derrotar, para llegar al final de esta escena.

Para pasar a la siguiente escena deberemos continuar andando al llegar al final del escenario.

En ese nivel nuestro personaje encontrará con distintos enemigos que una vez superados se terminará esa fase del del juego. La Figura ?? muestra un ejemplo de la escena 1.

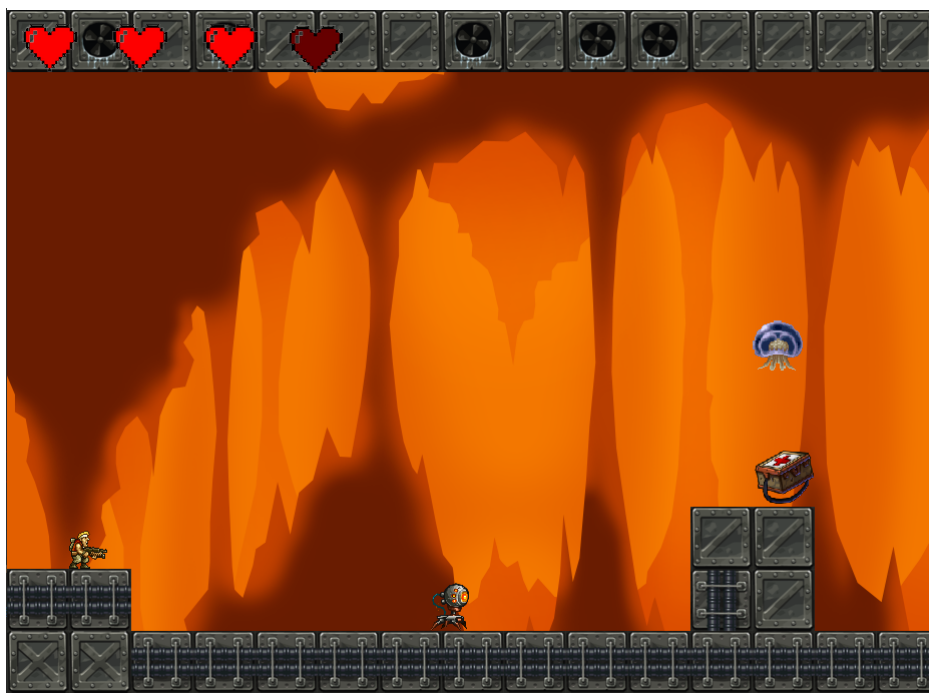


Figura 2.8: Ejemplo de la fase 1

Fase 2 - Almacén

En esta segunda escena el protagonista se encuentra con 3 vidas en el almacén, donde tendrá que recoger todas las herramientas (un total de 11) que se va a ir encontrando para completar con éxito esta etapa, también se encontrará con botiquines que le permitirán recuperar vidas, las cuales le podrán ser arrebatadas por los enemigos que entorpecerán su camino, tendrá que hacer frente a 7 tortugas, 6 spiderbots y 7 octopus. Si continúa andando al llegar al final del escenario, pasará a la escena 3. En la Figura ?? presenta un ejemplo de la misma.



Figura 2.9: Ejemplo de la fase 2

Fase 3 - Puente de mando de la nave

En la tercera y última escena el jugador aparece en el Puente de mando con 3 vidas, al igual que en las escenas anteriores, debe llegar al final del escenario con vida intentando destruir a los enemigos que se encuentre a su paso, un total de 20: 7 tortugas, 7 octopus y 6 spiderbots. Si consigue llegar con vida al final de esta escena habrá completado la aventura y llegará a la pantalla de Victoria. En la Figura ?? presenta un ejemplo de la misma.



Figura 2.10: Ejemplo de la fase 3

Créditos

Pantalla en la que se visualiza la lista de nombres de los desarrolladores y del product owner. Desde esta pantalla se permite al jugador volver al menú principal pulsando la tecla Esc. Ejemplo en la Figura ??.

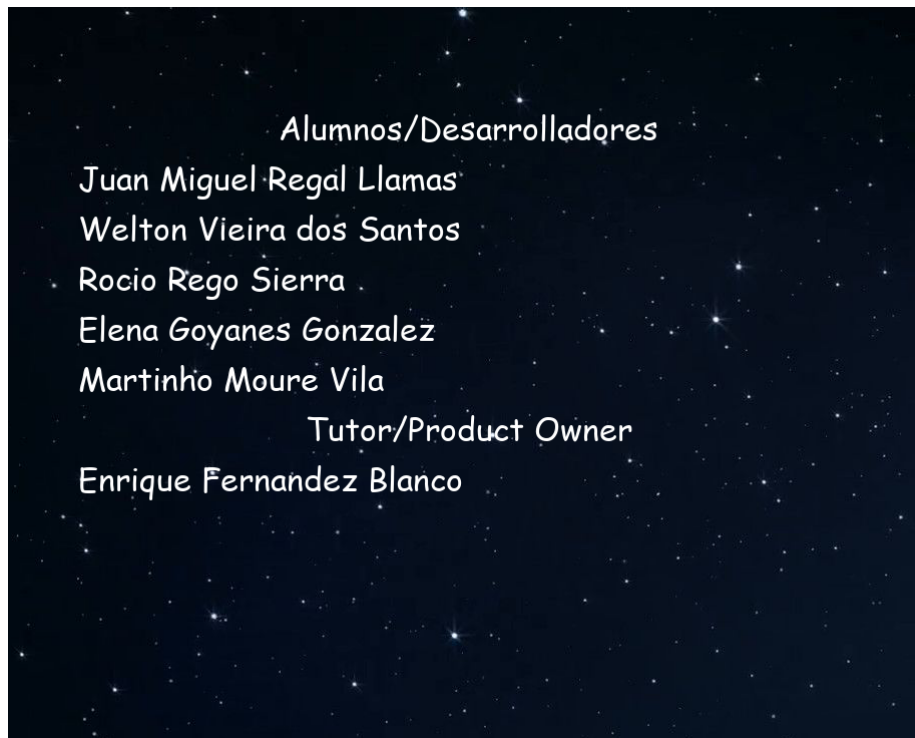


Figura 2.11: Ejemplo del apartado créditos del menú principal

Victoria

En esta pantalla se muestra un aviso de que el jugador ha conseguido ganar y terminar el juego. Para volver al menú principal se puede pulsar la tecla Esc. Ejemplo en la Figura ??.



Figura 2.12: Ejemplo de la escena de victoria

Derrota

Aquí se informa al jugador de que ha sido derrotado, tras haberse quedado sin vidas. Desde esta pantalla se puede volver al menú principal pulsando a tecla Esc. Ejemplo en la Figura ??.



Figura 2.13: Ejemplo de la escena de derrota

Pausa

Aquí se informa al jugador de que el juego sido pausado despues de precionar la tecla de escape (esc). Desde esta pantalla se puede volver al menú principal pulsando a tecla Esc o seleccionar la opción **Menu Principal**. Ejemplo en la Figura ??.



Figura 2.14: Ejemplo de pausar el juego

2.1.6 Aspectos destacables y detalles de su implementación

Creacción de las fases

Un aspecto del proyecto que cabe destacar es la forma en que se crean los escenarios y se posicionan los enemigos e items, esta parte se realiza a través de una arquitectura dirigida por datos, donde cada escenario está representado por una matriz de números en un fichero .txt, donde cada número representa un tipo de bloque diferente y los ceros que no hay ningún bloque.

Los tipos de bloque (“tiles”) están representados en ficheros json, con un array de

elementos, cada uno con un identificador que corresponde al valor numérico en la matriz. El nombre de la hoja de sprites que contiene el sprite del bloque y la posición en la hoja de sprites.

La colocación de los enemigos y los items es parecida:

Para cada enemigo y para cada item tenemos un fichero json formado por un array de elementos. Estos elementos contienen un nº identificador, la posición en el eje x y la posición en el eje y.

Después estos datos son leídos por las funciones:

“readMatrix” : Para leer a matriz que representa o escenario

“readPosition”: Para leer de las posiciones de los items y los enemigos

“ReadTileReferences”: para leer o json que define los bloques (“tiles”)

Dinámicas de ataques

Tanto para los disparos del personaje, como para los enemigos y los items, para que no se crearan y destruyeran objetos de forma que acabara pasando el recolector de basura provocando que se atasque el juego, todos estos elementos son creados al iniciarse el nivel, y al tener que ser destruidos son cambiados del grupo al que corresponden a un grupo que no se actualiza, por lo que se mantienen sus referencias de modo que el juego no se atasca borrandolos, pero tampoco se pierde tiempo de ejecución actualizándolos.

Para los disparos, creamos un array de un nº máximo de balas suficientes para que se puedan reutilizar sin que el jugador se entere, este array pertenece al personaje que las dispara y sus balas al inicializarse pertenecerán al grupo de balas que no se actualizan, así, cada vez que dispare llamará a una función que le reinicia todos los atributos a la bala y esta se añade al grupo de balas que sí se actualizan.

En el protagonista al disparar se calculará el ángulo de disparo del ratón respecto al protagonista.

3 Manual

3.1 Estructura de las escenas

La estructura del videojuego es la siguiente:

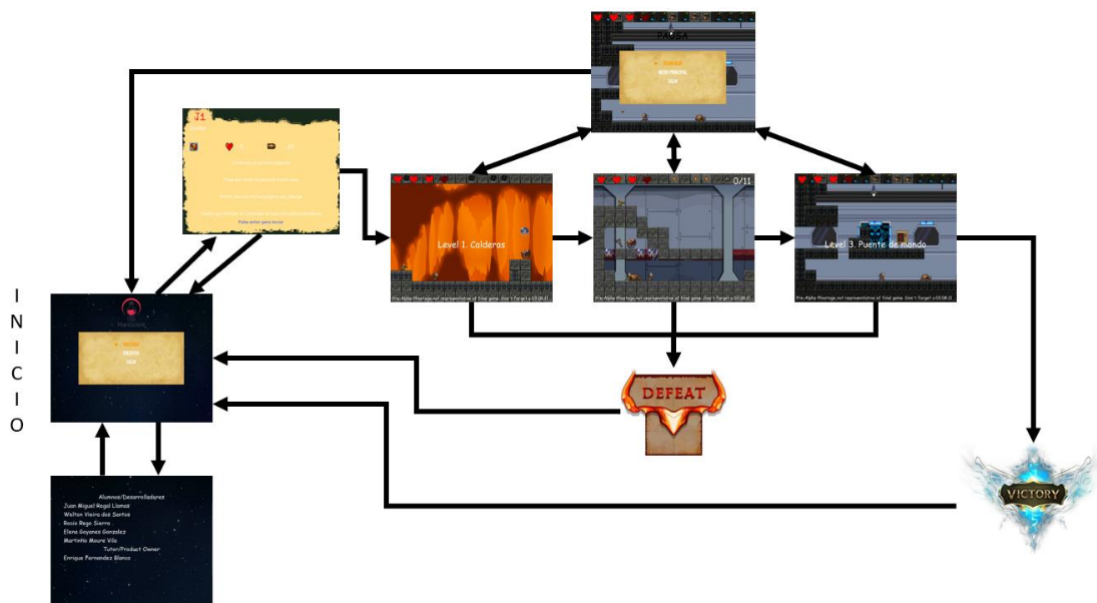


Figura 3.1: Estructura de las escenas del video juego

3.2 Controles

Las flechas del teclado permiten que el jugador se mueva por la pantalla en las direcciones de las flechas. Figura ??.

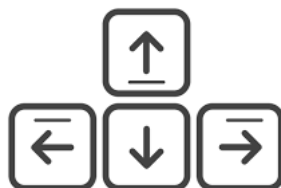


Figura 3.2: Teclas de movimiento del personaje

El ratón permite que el jugador apunte y dispare los proyectiles a sus enemigos con click izquierdo.



Figura 3.3: Raton y click izquierdo

La tecla escape permite al jugador acceder al menú de pausa



Figura 3.4: Tecla escape

3.3 Ejecución del juego

Para ejecutar el juego es preciso tener instaladas la librería pygame, tras eso, una vez posicionados dentro del directorio del proyecto, ejecutamos el juego con el comando:

```
> python3 main.py
```

3.4 Navegación del Menu Principal

Al iniciarse el juego aparece el menú de inicio con las siguientes opciones:

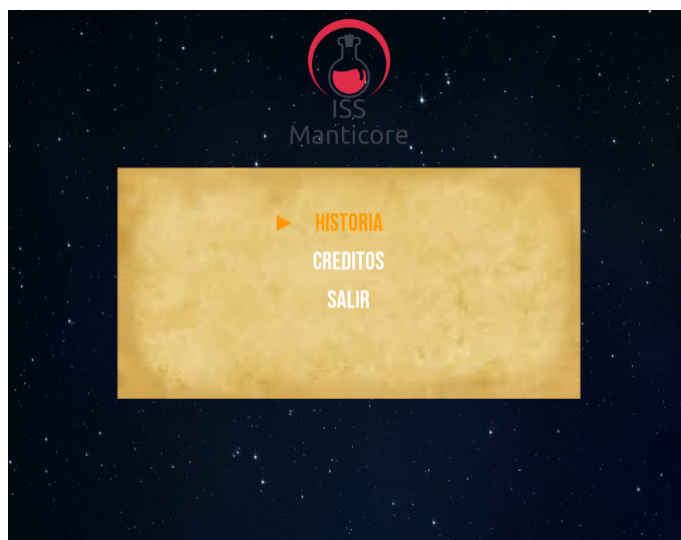


Figura 3.5: Ejemplo del menu principal

Para desplazarse entre las distintas opciones se utilizan las flechas del teclado y para seleccionarla se pulsa Enter.

A continuación, se detallan cada una de ellas:

- **Historia:** muestra la leyenda del juego, con la configuración inicial en la que aparece la imagen del personaje, la cantidad de vida con la que comienza el jugador y la cantidad de munición (para como ya se mencionó implementar en el juego un sistema de munición y armas complejo en el futuro)
- **Créditos:** muestra la lista de nombres de los desarrolladores y del product owner.
- **Salir:** cierra la ventana del juego.

Tanto desde la pantalla de la leyenda como desde los créditos se puede retroceder al menú pulsando la tecla Esc.

Cuando da comienzo la historia van apareciendo los diferentes niveles que el jugador debe superar, si este llega al final aparecerá un indicativo demostrando que ha finalizado

el juego. Si por el contrario el jugador pierde todas sus vidas por el camino aparecerá otro indicativo manifestando la derrota. Tras terminar el juego, tanto con éxito como si no, es necesario pulsar la tecla Esc para volver al menú principal.