BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

Ingeniería del Software

Sprint 1 Sunken

2021-2022

...

Titulación:

Grado en Informática de Gestión y Sistemas de Información

...

2º Curso(2º Cuatrimestre) ...

Página de GitHub

Página de Notion (actas)

Alan García, Álvaro Díez-Andino,
Danel Alonso, Gorka Crespo, Adrián López

18 de mayo de 2022

Índice

1. Intrducción

El juego a desarrollar ha sido hundir la flota. Un juego de mesa clásico el cual hemos llevado a digital.
Este juego se juega entre 2 jugadores; en nuestro caso la persona que juega y el propio ordenador. Cada jugador tiene su tablero en el que colocará 4 tipos distintos de barco: Portaviones, submarinos, destructores y fragatas. Estos barcos son distintos entre sí puesto que ocupan 4, 3, 2, 1 casillas respectivamente. El jugador colocará sus barcos a placer en la orientación que escoja para cada barco, guardando 1 casilla mínima de distancia entre barcos y sin que se salgan del tablero lógicamente. Una vez colocados los barcos, podremos poner escudos a 2 de nuestros barcos para protegerlos frente al primer disparo que caiga y cuando tengamos nuestro tablero a nuestro gusto pulsaremos en el botón "Iniciar partida" y el ordenador colocará sus barcos de manera aleatoria para comenzar la partida.

El jugador comienza disparando y tendrá 2 tipos de disparo: la bomba (disparo normal, ilimitado que afecta a una casilla), la "Bomba OneTap" (una bomba que en caso de tocar barco si tiene escudo lo rompe y si no tiene escudo hunde el barco directamente). Otra opción podría ser colocar el radar. Este se colocará de manera aleatoria. Si queremos cambiarlo de posición tendremos que pulsar en el botón "Recolocar" para que se ponga aleatoriamente en otra posición. Siempre que coloquemos o recoloquemos el radar, contará como un turno.

Al igual que el jugador, el ordenador dispone de las mismas opciones a la hora de efectuar su turno solo que este hace todo aleatoriamente (hemos añadido niveles de probabilidad para efectuar diferentes acciones). Una vez hundidos todos los barcos de un tablero la partida finalizará. Quedando como perdedor el jugador con todos los barcos hundidos y ganador el que ha conseguido hundir los barcos del rival.

En el transcurso de la partida, tendremos un botón que nos abrirá una nueva ventana, la cual será una tienda para poder comprar diferentes cosas a lo largo de la partida. Para ello cada jugador comenzará la partida con una cantidad de dinero que irá gastando a lo largo de la partida en las elecciones que haga. Así bien no se podrá conseguir dinero a lo largo de la partida.

2. Desarrollo del proyecto

En el desarrollo del proyecto se ha seguido la metodología SCRUM y se han dividido las tareas en tres entregas o sprints principales.

2.1. Sprint 1

Partimos de cero, solo con el enunciado que nos han facilitado. Tras leerlo varias veces comenzamos a distinguir clases y sus atributos para conseguir el primer diagrama de clases. Una vez diseñado este diagrama de clases inicial comenzamos a pensar los métodos que cada clase debe tener.

Por último comenzamos con el código programando dichos métodos y añadiendo más según avanzamos ya que eran necesarios. Para este primer sprint el objetivo era tener el juego funcional (colocar barcos, disparar, tocar y hundir barcos y que la partida acabe cuando tiene que acabar), presentar el diagrama de clases resultante, describir cómo funciona el observer (informe MVC) y el diagrama de secuencia del método que realiza el disparo.

2.2. Sprint 2

Continuamos el proyecto en este Sprint con 2 cosas a añadir: escudos y radar. Al tener que añadir estas dos nuevas funcionalidades el diagrama de clases iba a cambiar conteniendo más cosas para que funcione como queremos. Acordamos en colocar los escudos antes de comenzar la partida y que la ubicación del radar fuese aleatoria gastando turno cada vez que volvemos a colocarlo. Así pues aparte de añadir estas dos nuevas opciones corregimos algunas del Sprint1. Finalmente diseñamos el diagrama de secuencia para el método que usa el radar y en la reunión final del Sprint enseñamos que el juego funciona con estas dos nuevas opciones (radar y escudo), el "nuevo" diagrama de clases el cual contiene lo anterior más lo necesario para implementar el radar y el escudo y el diagrama de secuencia previamente mencionado. Cabe mencionar que a parte del escudo y el radar añadimos un nuevo tipo de bomba; la bomba "OneTap" , la cual si impacta un barco rompe el escudo en caso de que lo tenga y si no es el caso hunde el barco directamente.

2.3. Sprint 3

Para esta entrega final queda implantar la tienda, poder reparar barcos y pulir el juego. Para reparar los barcos hemos puesto un botón que al pulsarlo y después pulsar una casilla tocada la restaura. Si el barco está hundido este no se podrá reparar. Para la tienda, hemos hecho una pestaña diferente para verlo más claro, en ella podemos comprar reparaciones, bombas "OneTap" y radares para usarlos en nuestra partida.

Así pues no hemos hecho pruebas como tal. Lo que hemos hecho ha sido jugar y jugar para ver que todo funciona a nuestro gusto y cuando el juego daba error o algo no funcionaba como lo teníamos pensado íbamos al código para solucionarlo

3. Manual

Con los botones que se encuentran a la derecha de nuestro tablero, seleccionamos entre los 5 tipos de barco que queremos colocar.
Estos botones tienen un número a su derecha que nos indica los barcos de ese tipo que queden por colocar.
Además, podemos elegir de qué manera orientar nuestro barco en el tablero, eligiendo la orientación "Norte","Este","Oeste" o "Sur".
En el caso de que no queramos colocar manualmente los barcos, existe la opción de que se coloquen automáticamente con el botón "Colocar Barcos".
Una vez colocados, podemos poner un escudo a 3 de nuestros barcos para que estén más protegidos.
Cuando tengamos la configuración del tablero finalizada, podemos iniciar la partida pulsando el botón "Iniciar Partida".

Una vez comenzada la partida, el jugador tendrá el primer turno en el que puede decidir si quiere disparar o colocar un radar.
Si queremos utilizar una bomba, la seleccionamos en el panel y pinchamos en una casilla del tablero enemigo.
Al disparar, la casilla se convertirá en una de color azul si es agua o roja si se ha tocado un barco.
Si hundimos un barco, este se pondrá de color amarillo y será rodeado por agua automáticamente.
Además, disponemos de otro tipo de bomba llamada "OneTap", la cual rompe el escudo de un barco en caso de tenerlo, y si no lo tuviera hunde el barco directamente.

El uso del radar ,en cambio, se situará en una casilla aleatoria. Al pulsar el botón "Colocar radar", este se pondrá en una casilla al azar del tablero enemigo y se pasará un turno.
Cuando se usa un radar, se muestra con un radio de 1 casilla si hay agua o un barco en la casilla que hayamos seleccionado.
El número de radares disponibles se mostrará a la derecha del botón "Recolocar Radar".

Aparte de estas dos opciones, existe un botón "Tienda" en la que podremos comprar bombas o radares con el dinero inicial de la partida (el cual no puede

aumentar en ningún momento).

Una vez hayan sido hundidos todos los barcos de un tablero la partida finalizará.
Finalmente, se abrirá una pestaña nueva diciéndote si has ganado o has perdido.

Buena suerte!!

4. Conclusiones

Ha sido un trabajo largo e intenso que creemos que hemos llevado bien. Tras acabarlo y ver el resultado estamos satisfechos con él. Como grupo nos hemos complementado bastante bien a la hora de hacer reuniones para estar siempre todos y llevar todo más o menos en común.

Si nos ponemos a pensar en cosas que podríamos añadir, pensamos en implementar otro modo de juego para jugar entre dos jugadores. En este modo, 2 jugadores participan desde diferentes ordenadores en una misma red local mientras un ordenador hace de servidor y el otro se conecta a él para establecer conexión.

Otro extra que habíamos pensado es al tener un barco seleccionado y pasar el ratón por las casillas, en vez de ponerse de otro color solo la casilla sobre la que está el cursor, que se viese la silueta del barco pre-marcada y si no se puede poner el barco que la silueta no aparezca.

Y por último podríamos añadir algunas maneras de conseguir dinero a lo largo de la partida; por ejemplo, cada vez que hundes un barco.

Ya vistos los añadidos "técnicos" que mejoraríamos o implementaríamos, mencionar también que en un principio quisimos añadir alguna foto o animaciones a la hora de jugar ser más atractivo visualmente.

5. Diagramas

5.1. Diagrama de clases del modelo

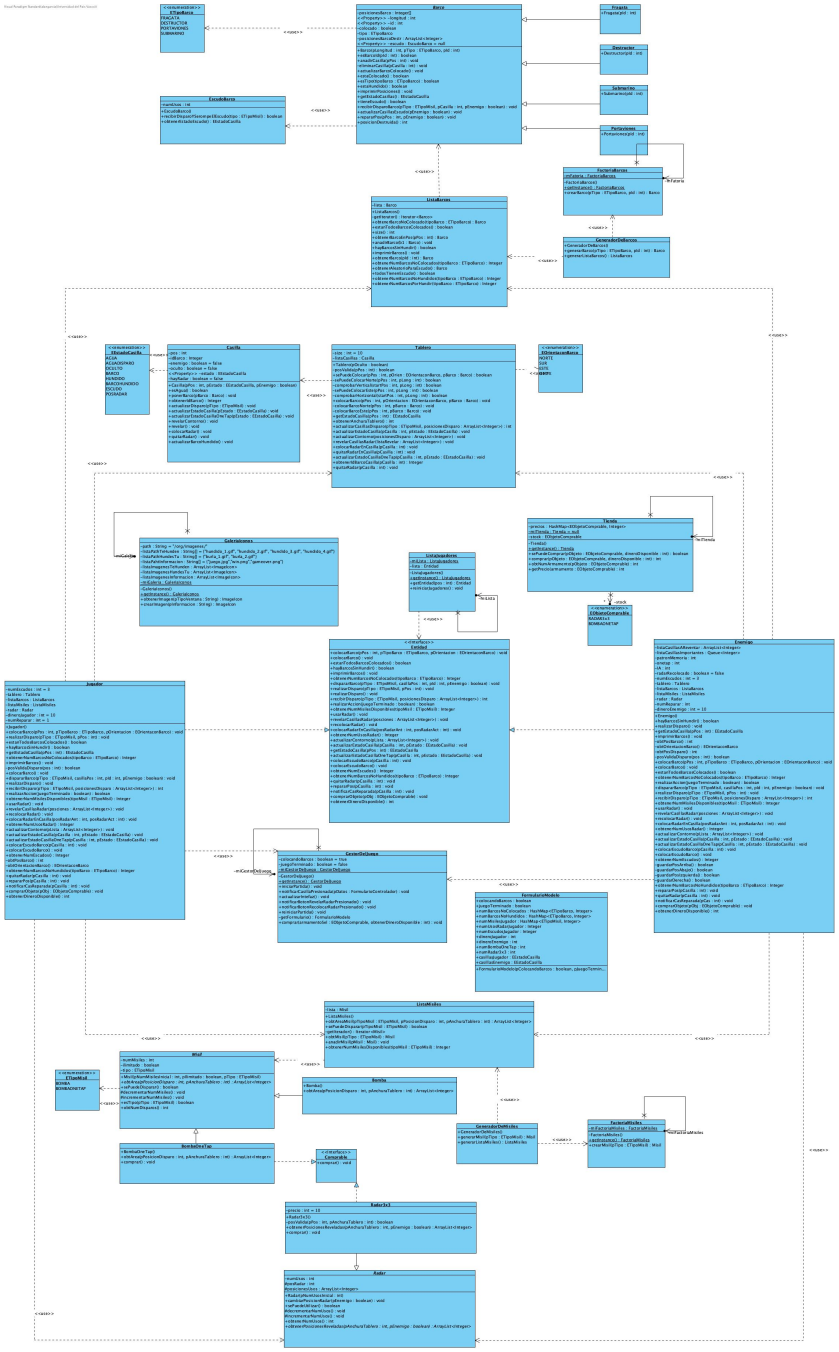

Figura 1: Diagrama de clases

5.2. Diagrama de secuencia de disparar Jugador

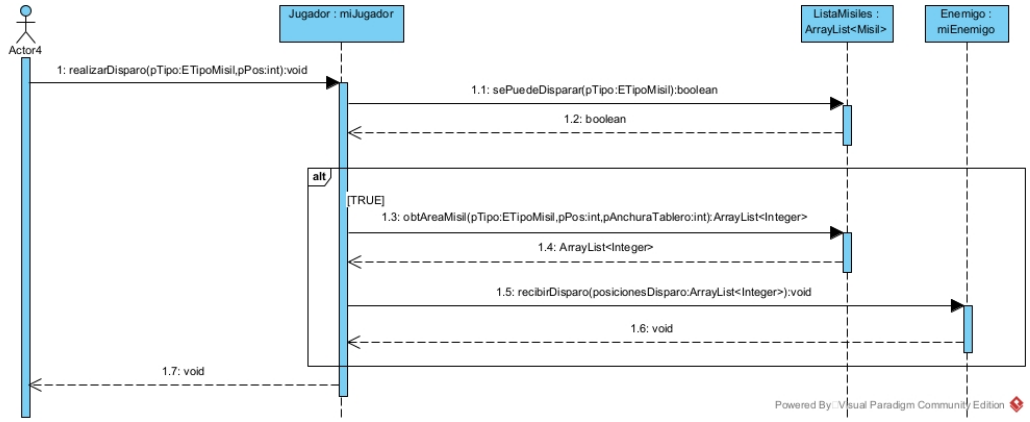

Figura 2: Realizar disparo jugador

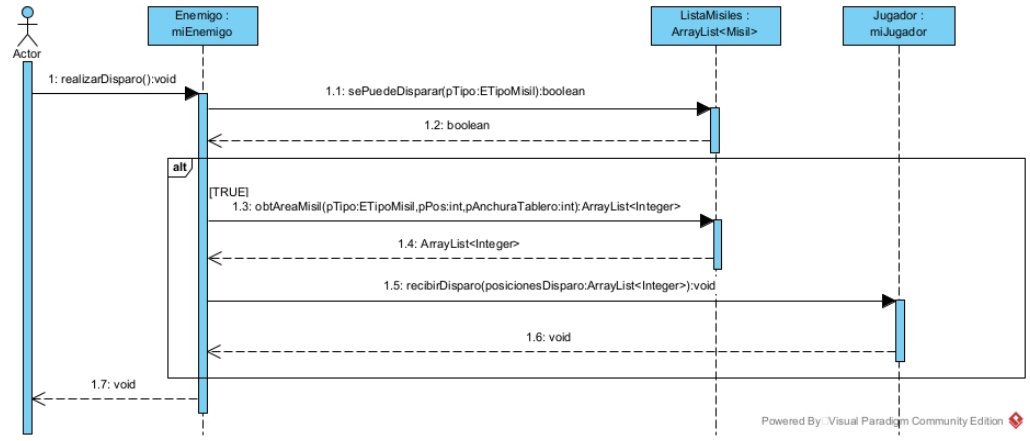

Figura 3: Realizar disparo enemigo

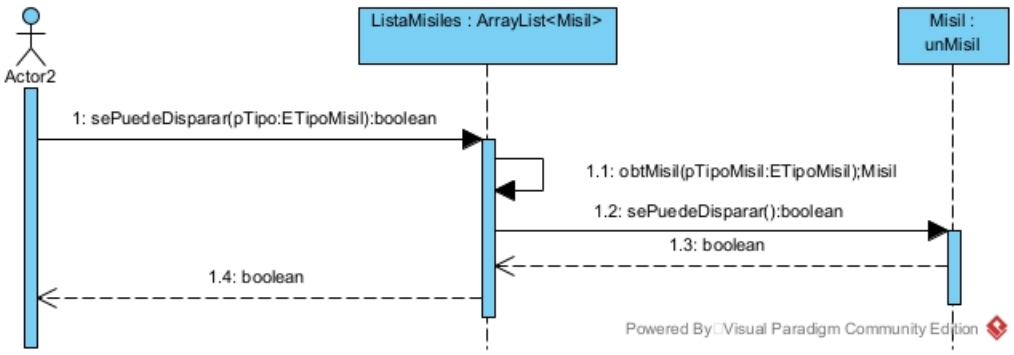

Figura 4: Se puede disparar de ListaMisiles

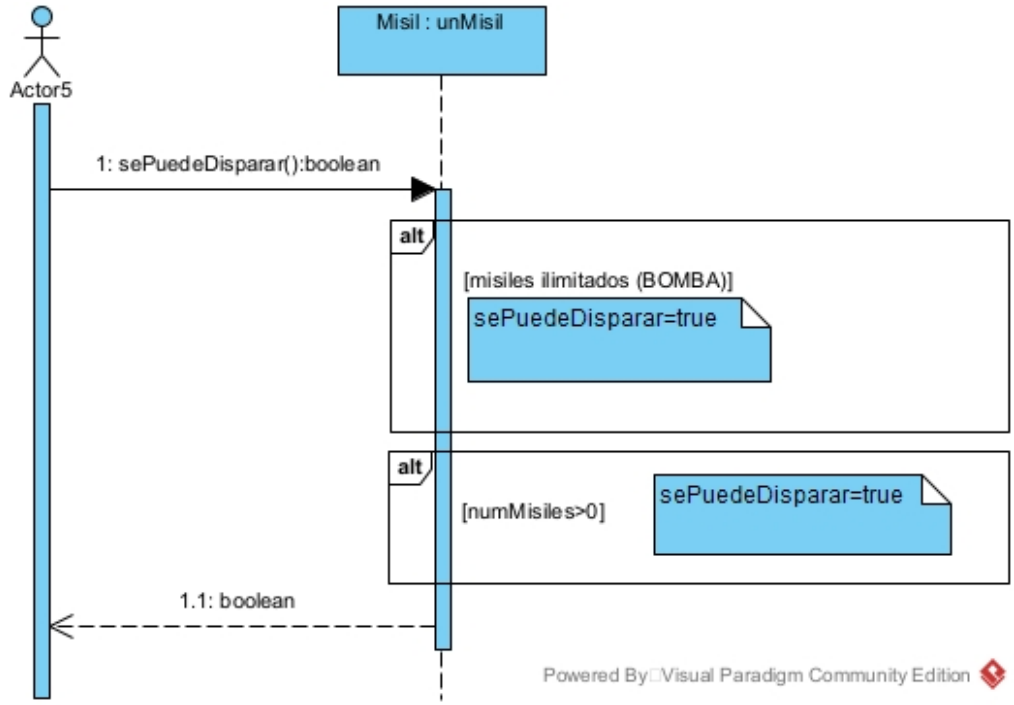

Figura 5: Se puede disparar de Misil

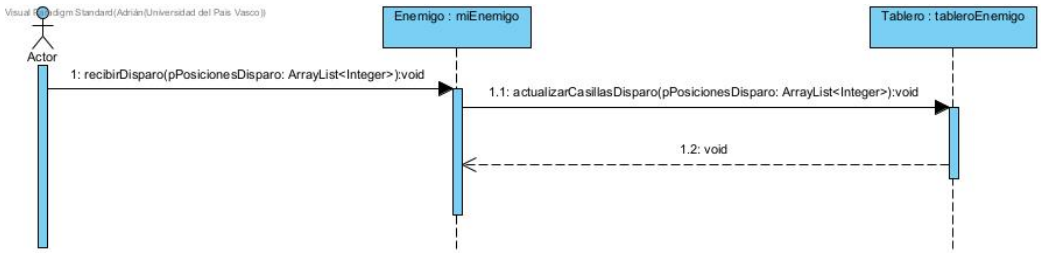

Figura 6: Recibir disparo enemigo

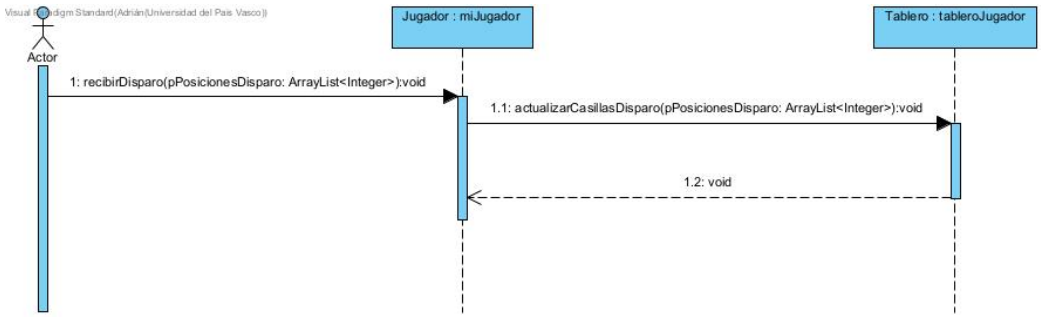

Figura 7: Recibir disparo jugador

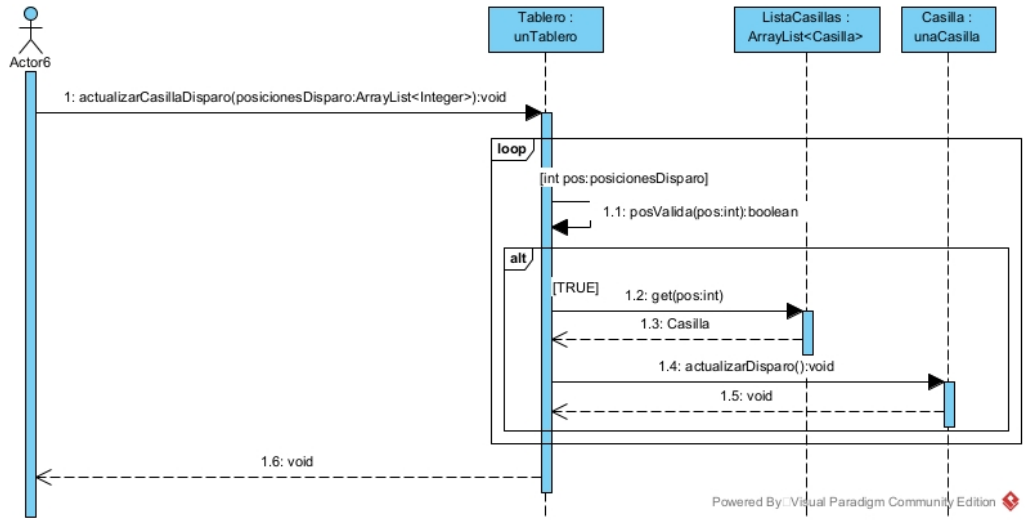

Figura 8: Actualizar casilla disparo

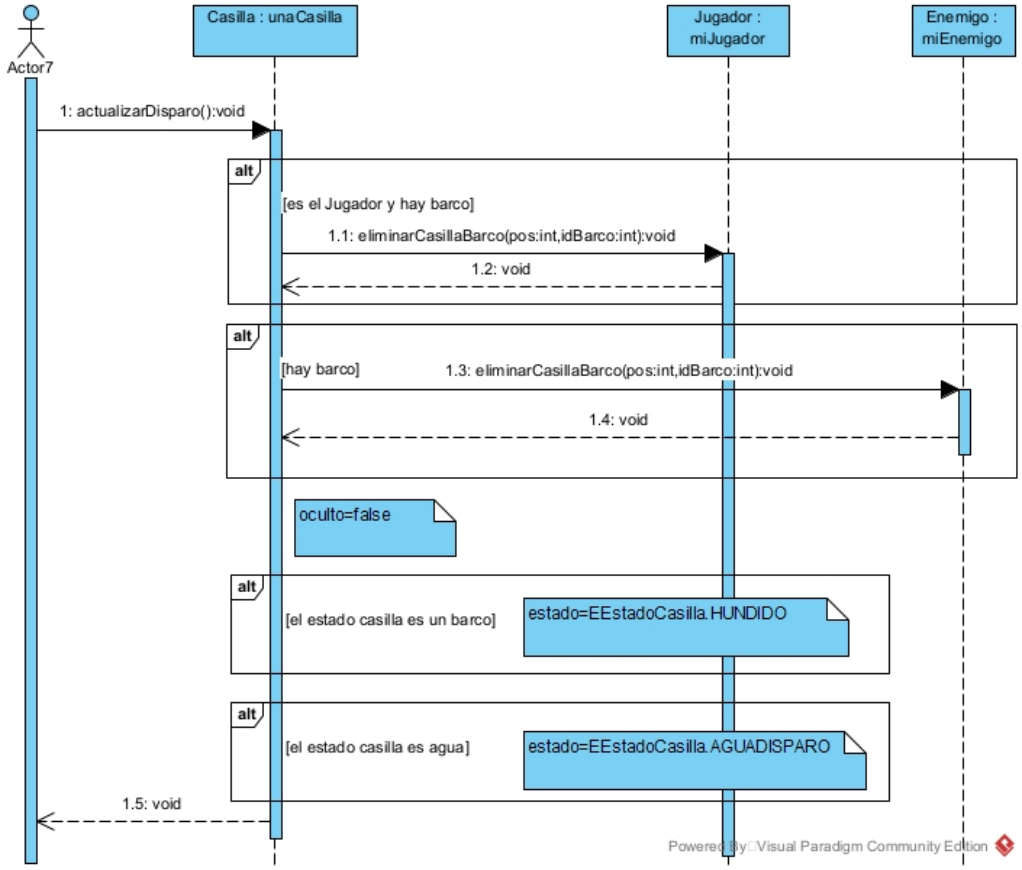

Figura 9: Actualizar casilla disparo

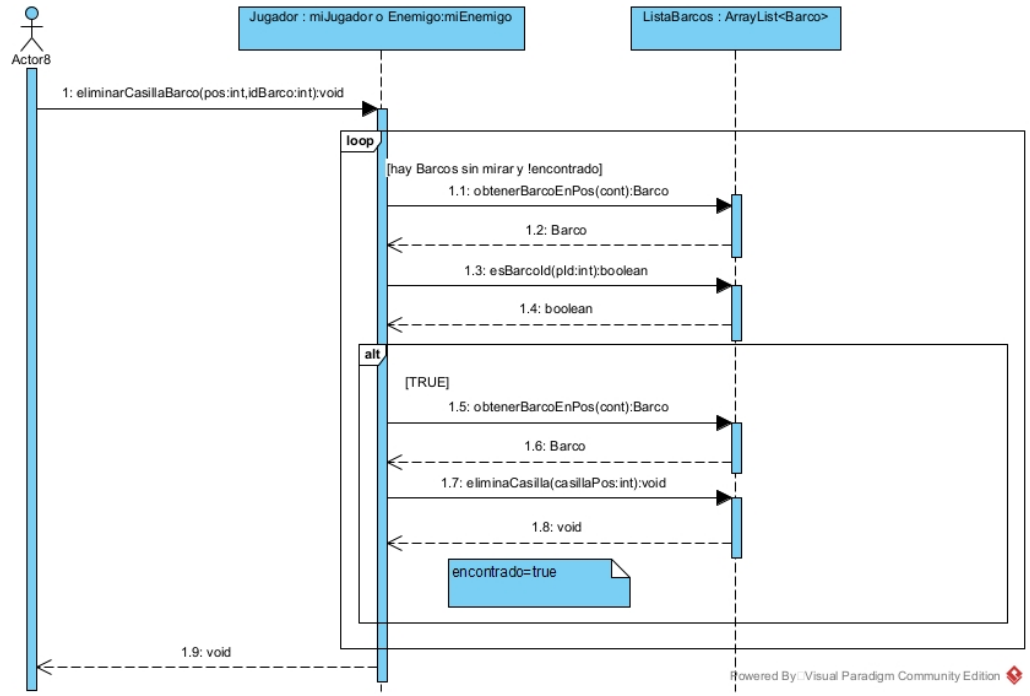

Figura 10: Eliminar casilla barco

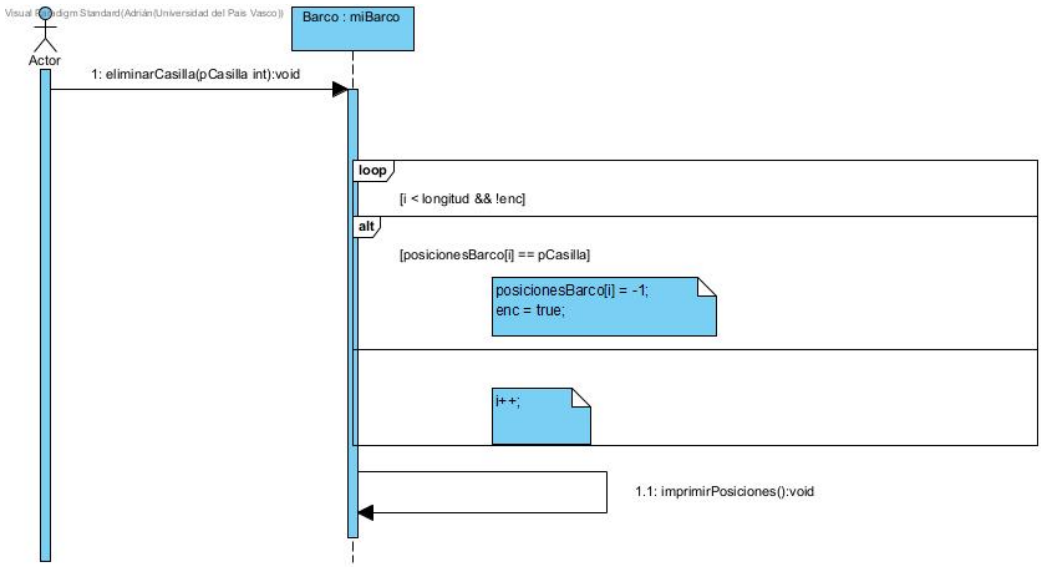

Figura 11: Eliminar casilla

5.3. Diagrama de secuencia de usarRadar Enemigo

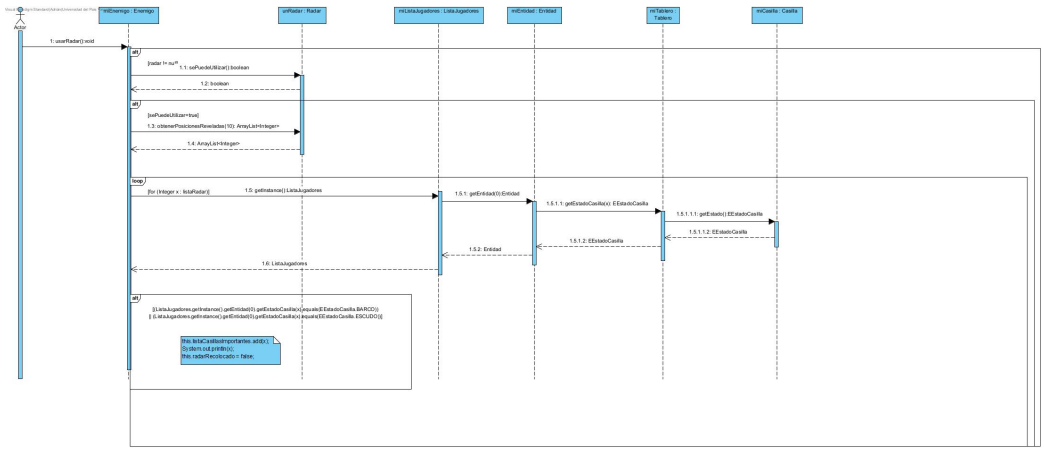

Figura 12: Usar Radar Enemigo

6. Actas de reunión

Las actas de reunión están incluidas como un pdf adjunto.

Primera reunión

Asistencia	Completa
Fecha Inicio	@April 26, 2022 5:00 PM
Fecha Fin	@April 26, 2022 7:00 PM
Lugar	Clase
Tags	Presencial

Conclusiones principales

La implementación de la tienda de armamento se realizará añadiendo atributos del dinero disponible tanto a jugador como enemigo y crearemos una clase denominada PreciosArmamento que contenga todos los precios de los objetos disponibles en dicha tienda. Las acciones de compra estarán contenidas dentro de jugador y enemigo porque los elementos del juego tienen unos métodos ya implementados que se encargan de controlar el número de usos de los armamentos y radares. Además, estas clases también contienen los atributos necesarios para controlar el número de escudos del jugador y enemigo, luego no tiene mucho sentido crear una nueva clase Tienda que realice estas acciones pues sólo añadiría complejidad al sistema.

En cuanto a las acciones relacionadas con reparar los barcos, ocurre algo parecido: pensamos que modificar las clases Barco, ListaBarcos y Entidad es mejor que añadir nuevas clases que aumenten la complejidad del asunto.

En relación a los patrones de diseño, no encontramos ningún caso en este sprint para añadir alguno.

Acuerdos tomados

- Crear una clase Tienda que recoja el Stock disponible y los precios
- Extender las clases actuales con los métodos necesarios
- Actualizar la interfaz (añadir barcos por hundir y otras opciones)

Tareas asignadas

Alan, Álvaro y Adrián → Implementación de métodos que tienen que ver con el funcionamiento de la Tienda.

Danel y Gorka → Implementación de la vista y los controladores que tienen que ver con el funcionamiento de la Tienda.

Segunda Reunión

Asistencia	Completa
Fecha Inicio	@May 3, 2022 3:00 PM
Fecha Fin	@May 18, 2022 5:00 PM
Lugar	Clase
Tags	Presencial

Conclusiones principales

La implementación de la tienda de armamento se realizará añadiendo atributos del dinero disponible tanto a jugador como enemigo y crearemos una clase denominada PreciosArmamento que contenga todos los precios de los objetos disponibles en dicha tienda. Las acciones de compra estarán contenidas dentro de jugador y enemigo porque los elementos del juego tienen unos métodos ya implementados que se encargan de controlar el número de usos de los armamentos y radares. Además, estas clases también contienen los atributos necesarios para controlar el número de escudos del jugador y enemigo, luego no tiene mucho sentido crear una nueva clase Tienda que realice estas acciones pues sólo añadiría complejidad al sistema.

En cuanto a las acciones relacionadas con reparar los barcos, ocurre algo parecido: pensamos que modificar las clases Barco, ListaBarcos y Entidad es mejor que añadir nuevas clases que aumenten la complejidad del asunto.

En relación a los patrones de diseño, no encontramos ningún caso en este sprint para añadir alguno.

Tareas asignadas

Álvaro → Implementar el método de reparar barcos para el Jugador.

Danel → Crear el controlador de la VentanaTienda, la VentanaTienda y introducir en la VentanaPrincipal el boton que repara el barco y hacer que para que funcione.

Tercera Reunión

Asistencia	Completa
Fecha Inicio	@May 4, 2022 5:00 PM
Fecha Fin	@May 4, 2022 7:00 PM
Lugar	Clase
Tags	Presencial

Conclusiones principales

La implementación de la tienda de armamento se realizará añadiendo atributos del dinero disponible tanto a jugador como enemigo y crearemos una clase denominada PreciosArmamento que contenga todos los precios de los objetos disponibles en dicha tienda. Las acciones de compra estarán contenidas dentro de jugador y enemigo porque los elementos del juego tienen unos métodos ya implementados que se encargan de controlar el número de usos de los armamentos y radares. Además, estas clases también contienen los atributos necesarios para controlar el número de escudos del jugador y enemigo, luego no tiene mucho sentido crear una nueva clase Tienda que realice estas acciones pues sólo añadiría complejidad al sistema.

En cuanto a las acciones relacionadas con reparar los barcos, ocurre algo parecido: pensamos que modificar las clases Barco, ListaBarcos y Entidad es mejor que añadir nuevas clases que aumenten la complejidad del asunto.

En relación a los patrones de diseño, no encontramos ningún caso en este sprint para añadir alguno.

Tareas asignadas

Álvaro → Implementar el método de reparar barcos para el Enemigo y arreglar pequeñas cosas del tablero.

Alan → Implementar el metodo de compra para que el Jugador compre armamentos.

Danel → Hacer que los cambios se visualicen en la VentanaTienda al ahora que el jugador compra.

Cuarta Reunión

Asistencia	Completa
Fecha Inicio	@May 8, 2022 10:00 AM
Fecha Fin	@May 8, 2022 12:00 PM
Lugar	Discord
Tags	Virtual

Conclusiones principales

La implementación de la tienda de armamento se realizará añadiendo atributos del dinero disponible tanto a jugador como enemigo y crearemos una clase denominada PreciosArmamento que contenga todos los precios de los objetos disponibles en dicha tienda. Las acciones de compra estarán contenidas dentro de jugador y enemigo porque los elementos del juego tienen unos métodos ya implementados que se encargan de controlar el número de usos de los armamentos y radares. Además, estas clases también contienen los atributos necesarios para controlar el número de escudos del jugador y enemigo, luego no tiene mucho sentido crear una nueva clase Tienda que realice estas acciones pues sólo añadiría complejidad al sistema.

En cuanto a las acciones relacionadas con reparar los barcos, ocurre algo parecido: pensamos que modificar las clases Barco, ListaBarcos y Entidad es mejor que añadir nuevas clases que aumenten la complejidad del asunto.

En relación a los patrones de diseño, no encontramos ningún caso en este sprint para añadir alguno.

Tareas asignadas

Álvaro → Arreglos en la inteligencia artificial del Enemigo, ya que a la hora de reparar un barco de Jugador, el Enemigo al disparar hacia cosas raras.

Alan → Arreglo de la estructura modelo-vista-controlador y patrón observer, ahora los datos se pasan a un formulario desde el modelo vista.

Danel → Implementar el método para que el Enemigo compre en la tienda y hacer que los cambios se visualicen en la VentanaTienda al ahora que el Enemigo compra.

Quinta Reunión

Asistencia	Adrián Alan Gorka Álvaro
Fecha Inicio	@May 16, 2022 10:00 AM
Fecha Fin	@May 16, 2022 12:00 AM
Lugar	Discord
Tags	Virtual

Conclusiones principales

La implementación de la tienda de armamento se realizará añadiendo atributos del dinero disponible tanto a jugador como enemigo y crearemos una clase denominada PreciosArmamento que contenga todos los precios de los objetos disponibles en dicha tienda. Las acciones de compra estarán contenidas dentro de jugador y enemigo porque los elementos del juego tienen unos métodos ya implementados que se encargan de controlar el número de usos de los armamentos y radares. Además, estas clases también contienen los atributos necesarios para controlar el número de escudos del jugador y enemigo, luego no tiene mucho sentido crear una nueva clase Tienda que realice estas acciones pues sólo añadiría complejidad al sistema.

En cuanto a las acciones relacionadas con reparar los barcos, ocurre algo parecido: pensamos que modificar las clases Barco, ListaBarcos y Entidad es mejor que añadir nuevas clases que aumenten la complejidad del asunto.

En relación a los patrones de diseño, no encontramos ningún caso en este sprint para añadir alguno.

Tareas asignadas

Alan → Arreglo de bugs al repararBarco (cuando presionabas reparar barco, podías elegir un misil y petaba) y añadir gifs cada vez que hundías o te hunden un barco.

Gorka → Hacer la documentación del proyecto.

Adrián y Álvaro → Agregar un botón a la VentanaMenu llamado manual, que antes de empezar la partida te explica el funcionamiento del juego.