Reunión con el cliente: Sprint 2

Nombre del grupo: Moby-Dick
Fecha: 27/04/2022
Comienzo de reunión: 17:05
Fin de reunión: 17:30
Lugar de reunión: Escuela de Ingeniería Bilbao UPV/EHU

Miembros del grupo presentes en la reunión:

Componente 1:	Javier Criado
Componente 2:	Xabier Gabiña
Componente 3:	Diego García
Componente 4:	Francisco González
Componente 5:	Estefanía Oñate
Componente 6:	Asier Santesteban

Requisitos a mostrar y explicar:

- ***Demo funcional con requisitos.***
- **Diagrama de clases**
- **Diagrama de Secuencia**
- **Planificación y reparto de tareas**

Reparto de exposición Sprint 1:

- **Demo:**
  - Javier Criado

- **Diagrama de Clases:**
  - Xabier Gabiña
  - Diego García

- **Diseño Diagrama de Secuencia:**
  - Francisco González
  - Estefanía Oñate

- **Documento Reparto de tareas:**
  - Asier Santesteban

Observaciones del cliente:

Al presentar la demo, aunque todo funcionase correctamente, el cliente nos ha remarcado que nuestro radar está mal implementado. Nuestro radar muestra si hay barcos en un radio inferior a 1 posición de la casilla seleccionada por el jugador, dejando dichas casillas despejadas hasta que se termine el juego. El cliente nos indica que el enunciado que nos cuenta lo siguiente:

- *El radar se sitúa en una posición del tablero de forma aleatoria.*
- *Se puede mover el radar a otra posición aleatoria tantas veces como lo permitan las consultas.*
- *Cuenta con un número de consultas que se decrementa con su uso, al acabarse estas el radar desaparece.*

Además el cliente nos dice que lo más ideal para la IA sería que, si tras usar el radar en el tablero del jugador esta encontrase una casilla ocupada por un barco, lo atacara.

En tanto a la observación del diagrama de clases el cliente se ha visto confundido por la relación ente IA y jugador, y tras darle nuestra explicación de seleccionarlos con un bool, siendo el jugador representado por 0 y la IA por 1, distinguimos las acciones de cada uno. La recomendación del cliente ha sido la de hacer una MAE lista debajo de jugador y que la IA heredase las características de esta, de esta forma si en el futuro se quisiera añadir un segundo jugador o más a la partida requeriría de menos cambios.
También ha mencionado una ligera corrección en que “radar” y “escudos” pertenezcan a la clase “armas”, ya que no se utilizan para atacar como tal, sino de utilidad para defenderse o saber la posición de barcos enemigos, para esto nos recomienda usar dos factorys, dejandola fuera de la parte de arma, ya que no necesita heredar sus atributos.
Para acabar con este diagrama, nos dice que la Dependencia entre el dinero que tiene el jugador y la tienda tiene que ser más clara.

Respecto al diagrama de secuencia lo que más le ha extrañado al cliente ha sido el uso de “for each” en vez de iteradores. Nos ha pedido cambiarlo en vista a que los iteradores tienen un potencial mayor a for each en tanto al uso y hacer cualquier cambio.
Finalmente tanto hace falta mostrar la funcionalidad correcta del radar en el diagrama de secuencia, siguiendo los pasos que nos especifica el enunciado.

Conclusiones:

En este segundo Sprint parece que la impresión del cliente sigue siendo buena. Aunque no todos los miembros del grupo han podido exponer su parte, dado que somos 6 miembros para 4 exposiciones, hemos sido capaces de resolver, discutir y defender nuestro proyecto en conjunto cuando el cliente nos ha expuesto sus dudas y observaciones. Cabe destacar que en esta segunda reunión han surgido más dudas y correcciones de diseño que en la anterior.

La demo funciona correctamente y al cliente le han gustado los detalles como los efectos de sonido e interfaz, sin embargo vamos a tener que corregir la funcionalidad del radar al que se especifica en el enunciado y siguiendo las pautas dadas por el cliente.

El Diagrama de Clases está completo y es funcional ahora que tiene las modificaciones requeridas por el cliente, sin embargo sería recomendable cambiar algunas decisiones de diseño por otras más preferidas por el cliente tal como se ha especificado en las observaciones.

El Diagrama de Secuencia ha dado dolores de cabeza por la falta de iteradores y algunas confusiones como saber si se manejaba la flota o el barco y va a necesitar una corrección que lo adapte a las nuevas peticiones del cliente.

La documentación con el reparto de tareas y estimación de horas cumple con su requisito y se va acumulando la información obtenida en las reuniones para corregir u observar los fallos que van surgiendo según avanza el proyecto.

Notas de la reunion:

DIAGRAMA DE CLASES:

- El radar no limpia limpia una zona hasta el fin de la partida, sino que va moviéndose de zona a zona aleatoriamente.
- Cuando la IA encuentra un barco a través del radar debería atacar al barco.
- Gestionar si es el turno de IA o jugador con una lista (MAE) que vaya debajo jugador, luego la IA heredaría. Sera necesario cambiar disparar y radar..
- Utilizar dos observers?
- Más claridad en la dependencia entre el dinero del jugador y la tienda.
- Meter radar y escudo fuera de la parte de arma porque la herencia de esa función no es necesaria para estas, ya que no son “armas” persé.

DIAGRAMA DE SECUENCIA:

- Utilizar iteradores en vez de for each. Se pueden usar para cualquier cosa.
- Mostrar en el diagrama el cambio con el radar que se muestra aleatoriamente.
- Repasar el diagrama ciñéndose a lo que pide el enunciado.

.