**Centro CTC**

**CEI**

**Batalla Naval**

**Obligatorio**

**Programación 1**

Integrantes:

Santiago Gutierrez, Tobias Nuñez, Facundo Cardoso

Profesor: Nicolas Podchi

**2023**

Índice

1- Descripcion .........................................................................................................

2- Especificacion de requerimiento ......................................................................

3- Analisis .................................................................................................................

4 - Logros ................................................................................................................

4.1- Requerimientos cumplidos...............................................................................

4.2- Requerimientos no cuplidos ............................................................................

5- Conclusiones finales ............................................................................................

* **Descripción**

Desarrolar un programa de consola Javascript que represente el juego "Batalla Naval". El juego será contra la máquina (El capitán Binary), debiendo la misma comportarse como el segundo jugador.

El tablero se compone de 100 casilleros para cada jugador, donde el eje vertical será representado por letras y el horizontal por números.

Al comenzar el juego, se le debe permitir al usuario posicionar sus barcos de manera horizontal o vertical a lo largo del tablero. Los mismos no podrán ser superpuestos unos encima de otros ni de manera diagonal. Tampoco podrán posicionarse barcos en casilleros adyacentes a naves ya posicionadas.

+ 1 portaviones - 5 casilleros

+ 1 acorazado - 4 casilleros

+ 2 submarinos - 3 casilleros

+ 3 destructores - 2 casilleros

Binary deberá posicionar su flota de manera aleatoria. Para posicionar ambas flotas se debe tener en cuenta que ninguna nave quede por fuera del rango de coordenadas disponibles. Una vez se encuentren posicionadas ambas flotas, se comenzarán a intercalar los turnos usuario/Binary para atacar a la flota rival.

Para atacar, el jugador debe indicar las coordenadas (letra - número).

Cuando unos de los jugadores acierta un golpe, el sistema deberá informarlo permitiendo distinguir si el golpe fué fallido, acertado o crítico. En caso de que el golpe sea fallido, se pasará el turno al otro jugador, pero en caso de que acierte (acertado o crítico) el mismo jugador seguirá atacando.

Los ataques de Binary serán aleatorios, pero cuando acierte un golpe, se deberá tener en cuenta el mismo para que en posteriores ataques se intente seguir dañando la misma nave del jugador hasta derribarla.

Se debe tener en cuenta que solamente se podrán atacar casillas que no se hayan atacado anteriormente. En caso de que el usuario esté atacando se mostrará en pantalla el tablero correspondiente a la flota de Binary y viceversa. En dicho tablero se deben indicar los golpes recibidos permitiendo identificar entre fallidos, acertados y naves hundidas.

Ganará el primer jugador que logre hundir la flota rival.

PARA EQUIPOS DE A 3:

Realizar un sistema de puntaje record donde al comienzo de cada partida se pida el nombre del jugador.

El puntaje de cada partida deberá guardarse en un histórico el cual podrá ser consultado por el usuario a demanda.

Dicho ranking debe mostrarse al final de cada partida indicando en que posición del ranking quedó el puntaje de la partida finalizada.

El puntaje del jugador para cada partida será la cantidad de ataques realizados. A menos cantidad de ataques, mejor posición en el ranking.

Se agregará al ranking el puntaje del jugador ganador de cada partida (Binary o usuario).

* **Especificación de requerimientos**

1. Desarrollar un programa de consola en Javascript para el juego "Batalla Naval".

2. El juego será contra la máquina, representada por el capitán Binary.

3. El tablero consta de 100 casilleros para cada jugador, con ejes verticales representados por letras y horizontales por números.

4. El usuario debe posicionar sus barcos de manera horizontal o vertical sin superponerse ni colocarse en diagonal.

5. No se pueden posicionar barcos adyacentes a naves ya posicionadas.

6. Los barcos y sus tamaños son: 1 portaviones (5 casilleros), 1 acorazado (4 casilleros), 2 submarinos (3 casilleros) y 3 destructores (2 casilleros).

7. Binary debe posicionar su flota de manera aleatoria.

8. Los turnos se intercalan entre el usuario y Binary para atacar a la flota rival.

9. Los jugadores atacan indicando las coordenadas (letra-número).

10. El sistema informa si el golpe fue fallido, acertado o crítico.

11. Si el golpe es fallido, se pasa el turno al otro jugador; si acierta, el mismo jugador sigue atacando.

12. Los ataques de Binary son aleatorios, pero una vez que acierta un golpe, intentará dañar la misma nave en posteriores ataques.

13. Solo se pueden atacar casillas no atacadas previamente.

14. Se muestra en pantalla el tablero correspondiente a la flota de Binary cuando el usuario ataca y viceversa, indicando los golpes recibidos y naves hundidas.

15. El juego termina cuando un jugador logra hundir la flota rival.

16. Para equipos de 3 personas:

- Se registra el nombre del jugador al comienzo de cada partida.

- El puntaje de cada partida se guarda en un histórico.

- El jugador con menos ataques tiene mejor posición en el ranking.

- Se muestra el ranking al final de cada partida, indicando la posición del puntaje obtenido.

- Se agrega el puntaje del ganador (Binary o usuario) al ranking.

* **Análisis**

Para resolver los requerimientos detallados anteriormente y desarrollar el juego de Batalla Naval, seguimos los siguientes pasos generales:

1. Comenzamos diseñando el tablero definiendo la estructura del tablero y cómo se representarán las coordenadas. Usando una matriz

2. Posicionamos los barcos del jugador, que permite al usuario posicionar sus barcos en el tablero. Asegúrate de verificar las reglas de posicionamiento, como evitar superposiciones y colocación diagonal, así como la distancia mínima de otros barcos.

3. Posicionamos los barcos de Binary, implementamos la lógica para que Binary posicione sus barcos de manera aleatoria en el tablero, asegurando de que se cumplan las reglas de posicionamiento.

4. Juego principal: Inicia el ciclo principal del juego, alternando los turnos entre el usuario y Binary para atacar.

5. Ataques del jugador: Permite al jugador ingresar las coordenadas para realizar un ataque. Verifica si el ataque es válido y determina si fue fallido, acertado o crítico. Actualiza el tablero y muestra la información relevante al jugador.

6. Ataques de Binary: Binary realiza ataques aleatorios. Asegurando de que Binary no ataque casillas previamente atacadas y considera los aciertos anteriores para intentar dañar la misma nave del jugador.

7. Verificación de fin de juego: Después de cada ataque, verifica si un jugador ha hundido la flota rival. Si es así, termina el juego y muestra el mensaje de victoria correspondiente.

8. Puntaje y ranking: Registramos el nombre del jugador al inicio de cada partida y guarda el puntaje. Calcula el puntaje de cada partida como la cantidad de ataques realizados. Al final de cada partida, muestra el ranking actualizado y la posición del puntaje obtenido.

9. Consulta de ranking: Implementamos la funcionalidad para que el usuario pueda consultar el ranking histórico en cualquier momento.

* **Logros**

1. Desarrollo exitoso de un juego de Batalla Naval en Javascript.

2. Implementación de la mecánica de posicionamiento estratégico de barcos.

3. Alternancia de turnos entre el jugador y la IA (capitán Binary) para realizar ataques.

4. Verificación de los ataques y notificación de resultados (fallido, acertado).

5. Representación gráfica en pantalla del tablero enemigo y los golpes recibidos.

6. Finalización del juego cuando se hunde la flota enemiga.

* **Requerimientos cumplidos**

1. Se ha desarrollado un programa de consola en Javascript que representa el juego "Batalla Naval".

2. El juego es contra la máquina, representada por el capitán Binary.

3. El tablero se compone de 100 casilleros para cada jugador, con ejes verticales representados por letras y horizontales por números.

4. El usuario puede posicionar sus barcos de manera horizontal o vertical a lo largo del tablero.

5. Se han implementado las restricciones de posicionamiento de barcos, como evitar superposiciones, colocación diagonal y casillas adyacentes a naves ya posicionadas.

6. Binary posiciona su flota de manera aleatoria sin incumplir las restricciones de posicionamiento.

7. Se han implementado los barcos y sus tamaños: 1 portaviones (5 casilleros), 1 acorazado (4 casilleros), 2 submarinos (3 casilleros) y 3 destructores (2 casilleros).

8. Se han alternado los turnos entre el usuario y Binary para atacar a la flota rival.

9. El jugador puede atacar indicando las coordenadas (letra-número) y se verifica si el golpe fue fallido, acertado o crítico.

10. Los ataques de Binary son aleatorios y se tiene en cuenta el daño a las naves del jugador.

11. Solo se pueden atacar casillas no atacadas previamente.

12. Se muestra en pantalla el tablero correspondiente a la flota de Binary cuando el usuario ataca y viceversa, mostrando los golpes recibidos y naves hundidas.

13. El juego termina cuando un jugador logra hundir la flota rival.

* **Requerimientos no cumplidos**

Listado de requerimientos que no se han cumplido en el proyecto:

1. No se implementó la funcionalidad para guardar el puntaje de cada partida en un histórico o ranking.

2. No se implementó la opción para que el usuario pueda consultar el ranking histórico en cualquier momento.

3. No se implementó el golpe critico.

* **Conclusiones finales**

Santiago Gutierrez:

El proyecto ha sido completado con éxito, brindando una experiencia de juego estratégica y entretenida. Se han implementado las reglas del juego, como la verificación del posicionamiento de los barcos, la validez de los ataques y la notificación de los resultados de los golpes, ya sean fallidos o acertados. Además, se muestra en pantalla el tablero correspondiente a la flota enemiga, lo que permite al jugador visualizar los golpes recibidos y las naves hundidas.

Aunque no se cumplieron todos los requerimientos como la funcionalidad de guardar el puntaje, la consulta del ranking y el golpe critico, el proyecto ofrece una experiencia de juego satisfactoria y entretenida.

Tobias Nuñez:

Facundo Cardoso: