UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Programación II
Proyecto Final
Pregrado
2021-II
Profesor: Estanislao Contreras

1. Proyecto Final

Implementar el juego Battleship el cual involucra la participación de dos equipos.

Tableros

El programa deberá mostrar dos tableros que representan una zona diferente de mar abierto: la propia y la contraria.

- En el primer tablero, el equipo coloca su flota y registra los tiros del oponente.

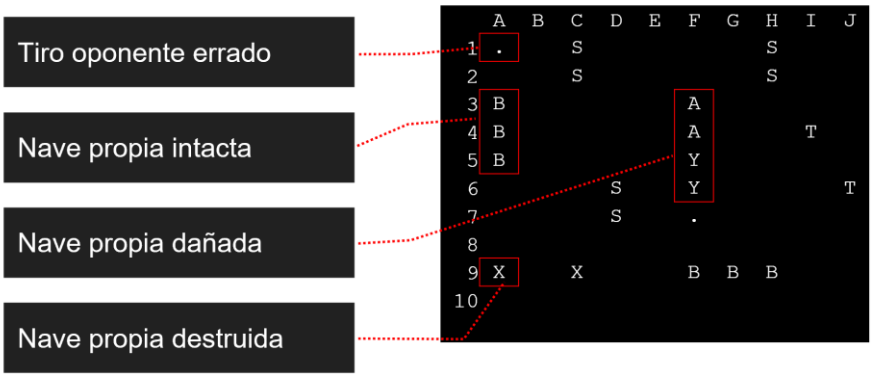


Fig. 1: Representación sugerida de la zona de mar propia

- En el segundo tablero se registran los tiros propios contra el otro equipo, diferenciando los impactos y los que dan al agua. Con esta información se deduce la posición de los barcos del contrincante.

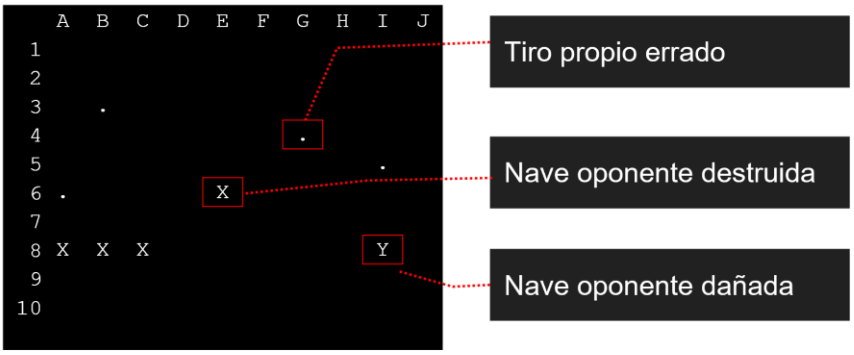


Fig. 2: Representación sugerida de la zona de mar del contrincante

Flota

Al comenzar, cada equipo posiciona su flota en el primer tablero, de forma secreta, invisible al oponente. Para representar una nave se ocupa casillas de forma horizontal o vertical:

- Aircfratf Carrier (A) → 4 casillas contiguas
- Battlecrucier (B) → 3 casillas contiguas
- Submarine (S) → 2 casillas contiguas
- Torpedo boat (T) → 1 casilla

Una flota está compuesta:

- 1 Aircfratf Carrier
- 2 Battlecruciers
- 3 Submarines
- 4 Torpedo boat

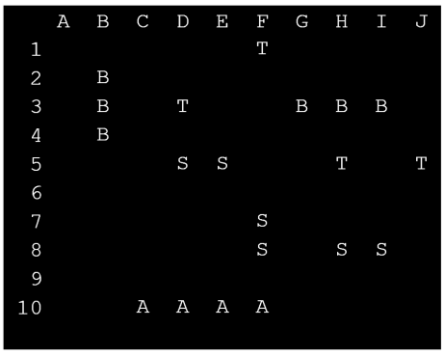


Fig. 3: Representación sugerida de una flota posicionada

Desarrollo del Juego

Una vez posicionada las flotas, se inicia una serie de rondas. En cada ronda, el programa de cada equipo dispara hacia la flota de su oponente indicando una posición (las coordenadas de una casilla), la que registra en el segundo tablero.

- Si esa posición es ocupada por parte de una nave contraria, la respuesta que recibirá es DAMAGED (Nave Dañada)
- Si con ese disparo todas las partes de la nave quedaron dañadas, la respuesta que recibirá es DESTROYED (Nave Destruida)
- El equipo que ha tocado una nave en su anterior jugada volverá a disparar hasta que falle.
- Si la posición indicada no corresponde a una parte de alguna nave, la respuesta que recibirá es FAILED (Tiro Fallado)

Cada equipo referenciará en el segundo tablero, de diferente manera y a su conveniencia, los disparos que han caído sobre una nave oponente y los que han caído al mar.

Fin del Juego

El juego debe terminar con un equipo ganador o en empate:

- **Ganador**: el equipo que destruya primero todas las naves de su oponente será el vencedor, en caso de que el participante que comenzó la partida hunda en su última jugada el último barco de su oponente que quedaba a flote, el otro participante tiene derecho a una última posibilidad para alcanzar el empate, a un último disparo que también le permita terminar de hundir la flota contraria, lo que supondría un empate.
- **Empate**: si bien lo habitual es continuar el juego hasta que haya un ganador, el empate también puede alcanzarse si, tras haber disparado cada jugador una cantidad máxima de tiros predeterminada (100 tiros), ambos equipos han acertado en igual número de casillas contrarias.

Detalles técnicos para el programa

- El mecanismo de comunicación entre los programas será a través de archivos de texto, los cuales se generarán en una carpeta compartida a los equipos que compiten.
- Los archivos con los comandos se deberán llamar FirstPlayer.in y SecondPlayer.in
- Los archivos con los resultados de los comandos se llamarán FirstPlayer.out y SecondPlayer.out
- Los comandos y sus respectivos resultados son los siguientes:

No	Tipo	Comando	Descripción
1	IN	HANDSHAKE=<Equipo>	Registrar equipo en el juego. Por ejemplo: HANDSHAKE=Royal Navy
	OUT	ACCEPTED TOKEN=<Token>	Indicativo si fue aceptado o rechazado y token que servirá para las siguientes comunicaciones
2	IN	TOKEN=<Token> PLACE_FLEET=<Posición>	Posicionar nave: Nave-Coordenadas-Orientación Por ejemplo: PLACE_FLEET=A-A1-H
	OUT	ACCEPTED MESSAGE=<Mensaje>	Indicativo si fue aceptado o rechazado más un mensaje de que acciones puede seguir haciendo.
3	IN	TOKEN=<Token> ATTACK=<Posición>	Disparar a la posición ColumnaFila Por ejemplo: ATTACK=B3
	OUT	ACCEPTED MESSAGE=FAILED	Indicativo si fue aceptado o rechazado más el resultado del ataque: ***FAILED***: Tiro cayo en el agua ***DAMAGED***: Tiro impacto parte de la nave ***DESTROYED***: Tiro hundió la nave

Detalles técnicos para el programa

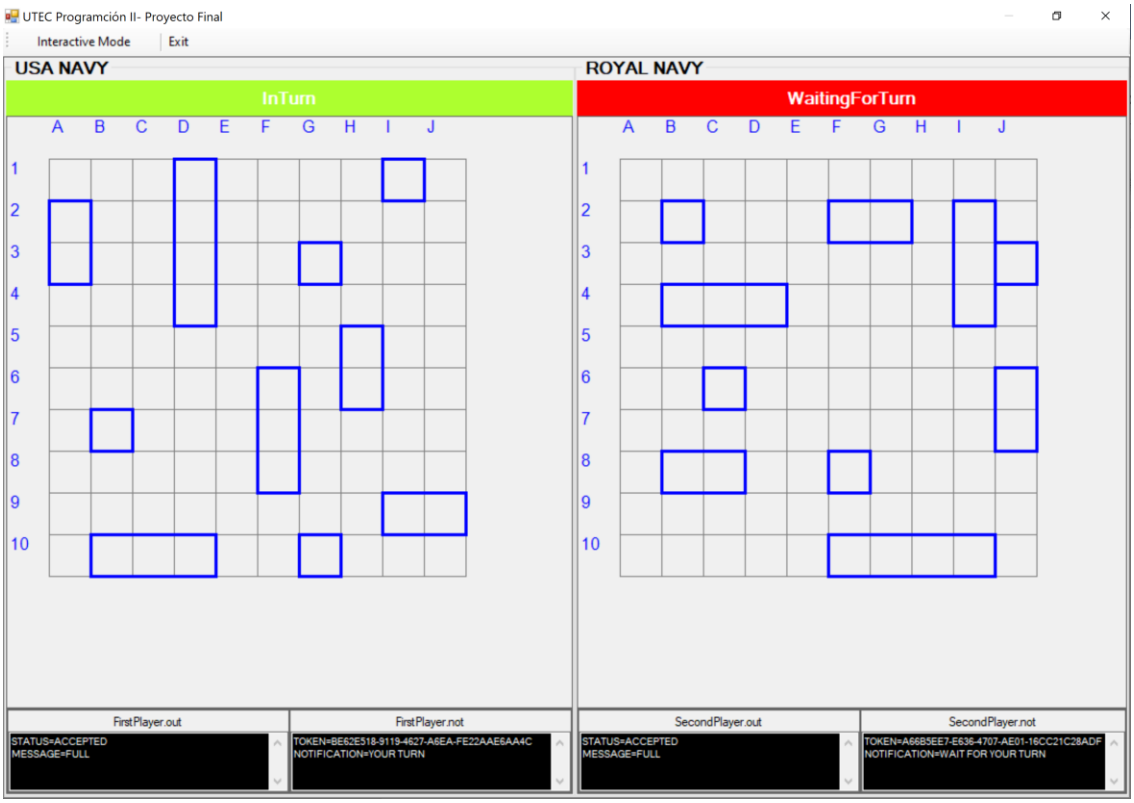
- El programa deberá tener la inteligencia suficiente para determinar la mejor estrategia de ataque en base a los resultados de cada disparo realizado.
- El programa deberá funcionar de forma autónoma (escenario esperado) o con poca intervención humana (peor escenario)
- No es necesario contar con una interfaz gráfica para mostrar el desarrollo del juego, suficiente con mostrar las 2 zonas.

```
Zona Contraria
   A  B  C  D  E  F  G  H  I  J
 1
 2
 3     .
 4                       .
 5                             .
 6  .           X
 7
 8                             Y
 9
10

Zona Propia
   A  B  C  D  E  F  G  H  I  J
 1  .     S                 S
 2        S                 S
 3  B                 A
 4  B                 A        T
 5  B                 Y
 6           S        Y           T
 7           S        .
 8
 9  X     X           B  B  B
10

>ATTACK=I9
```

- La competición será controlada por otra aplicación, que será provista por el profesor del curso, el cual se podrá usar para entrenar a la aplicación que cada equipo debe implementar.



2. Rúbrica:

Criterio	EXCELENTE	ADECUADO	MÍNIMO	INSUFICIENTE
Desarrollo de software	Diseña y elabora el software para lograr una solucion adecuada al problema planteado. El software debe ser ordenado, claro y optimo. (12 p.)	Diseña y elabora el software para lograr una solucion adecuada al problema planteado. El software es solo funcionable. (8 p.)	Diseña el software para lograr una solucion adecuada al problema planteado. El software no se concluye adecuadamente. (4 p.)	No logra el diseño ni la implementacion correcta del software. (2 p.)
Presentación escrita	El informe contiene las secciones de Antecedentes, Fundamento Teórico, Metodos y Desarrollo y Conclusiones. Estas últimas, adecuadamente formuladas. (4 p)	El informe contiene las secciones de Antecedentes, Fundamento Teórico, Metodos y Desarrollo, pero no pone enfasis en las conclusiones. (3 p)	El informe contiene menos de la mitad de las secciones estipuladas, incluyendo conclusiones. (2 p)	El informe contiene menos de la mitad de las secciones estipuladas, sin incluir conclusiones. (1 p)
Presentación oral	El alumno presenta el proyecto en forma adecuada y responde a las preguntas del profesor en forma logica y coherente. (4 p)	El alumno presenta el proyecto en forma adecuada y pero no responde a todas las preguntas del profesor en forma logica y coherente. (2 p)	El alumno no presenta el proyecto en forma adecuada pero responde a las preguntas del profesor en forma logica y coherente. (1 p)	El alumno no presenta el proyecto en forma adecuada ni responde a las preguntas del profesor en forma logica y coherente. O no se presenta a la presentacion oral. (0 p)