

Objetivo

El objetivo es crear un programa C que permita a un usuario jugar una partida contra la PC a la Batalla Naval.

Detalles

Los tableros del usuario y del sistema consisten en una matriz de 100 casilleros:

	0	1	2	3	4	5	6	7	8	9
A										
B										
C										
D										
E										
F										
G										
H										
I										
J										

Los tableros estarán cargados con **barcos** cuya cantidad y tipo se indican en la siguiente tabla, la cual llamaremos **flota**:

Tipo de Barco	Cantidad Casilleros	Cantidad Barcos
Portaaviones	5	1
Destructor	4	2
Acorazado	3	2
Buque	2	3
Submarino	1	4

Tabla 1: Flota

Los barcos se pueden disponer **horizontalmente o verticalmente, no así en diagonal.**

Por ejemplo, una **distribución posible** sería esta, donde “~” indica “agua”.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[A]	~	P	~	~	~	~	~	S	~	~
[B]	~	P	~	~	~	~	~	~	~	~
[C]	~	P	~	B	B	~	~	B	~	A
[D]	~	P	~	~	~	~	~	B	~	A
[E]	~	P	~	~	A	A	A	~	~	A
[F]	~	~	S	~	~	~	~	~	~	~
[G]	~	~	~	D	~	~	D	D	D	D
[H]	~	~	~	D	~	~	~	~	~	~
[I]	S	~	~	D	~	~	~	~	~	~
[J]	~	~	~	D	~	B	B	~	~	S

Figura 1

Al **usuario** se le ofrecerá las siguientes **alternativas para el proceso de carga** de la flota:

- **carga automática**, misma función a emplear para la carga de la flota del sistema
- **carga manual**, donde podrá introducir su flota e ir observando el estado a medida que va cargando cada uno de los barcos.
- **carga desde un archivo de texto**, el cual permitirá que el usuario exprese, en un formato a definir por el programador, como desea que se distribuya su flota.

El **sistema** realizará un **proceso de carga automático o bien desde archivo de texto de su tablero** estableciendo los barcos sin superponerlos.

Luego del proceso de carga, tanto para la flota del sistema como para el del usuario, **deberá ejecutarse una rutina que compruebe la validez de los tableros** con el fin de que el juego se desarrolle con tableros correctamente confeccionados.

El usuario deberá ver por pantalla la siguiente información:

- su tablero (como fue cargado, similar al de la **Figura 1**)
- en el mismo tablero, las jugadas del sistema, destacando de alguna manera cada casillero que el sistema vaya descubriendo
- cantidad de barcos del sistema que han sido destruidos por el usuario
- status de juego, el primero que consigue impactar a todos los barcos por completo gana la partida

Por supuesto, el usuario no debe ver el tablero del sistema.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[A]	~	P	~	~	~	~	~	S	~	~
[B]	~	P	~	~	~	~	~	~	~	~
[C]	~	P	~	B	B	~	~	B	~	A
[D]	~	P	~	~	~	~	~	B	~	A
[E]	~	P	~	~	A	A	A	~	~	A
[F]	~	~	S	~	~	~	~	~	~	~
[G]	~	~	~	D	~	~	D	D	D	D
[H]	~	~	~	D	~	~	~	~	~	~
[I]	S	~	~	D	~	~	~	~	~	~
[J]	~	~	~	D	~	B	B	~	~	S

Aquí se optó por destacar con color de fondo las jugadas del sistema (celeste para agua y naranja para los impactos) El programador puede elegir otras maneras de destacar las jugadas del sistema.

Ud. ha destruido:

- Portaaviones 0 de 1
- destructores 1 de 2
- Acorazados 2 de 2
- Buques 0 de 3
- Submarinos 3 de 4

Status del juego:

Usuario 8 impactos de 29
Sistema 5 impactos de 29
Jugada 14 de 100 jugadas posibles

Proceso de juego

El usuario es el que comienza la partida, indicando fila y columna a descubrir del tablero del sistema.

Debe verificarse que el usuario ingresa una posición (fila y columna) válida, indicado con un mensaje similar a este en caso contrario:

"Fila o columna inválida, ingrese nuevamente..."

Si el casillero que el usuario propone ya fue descubierto en una jugada anterior, deberá ser notificada tal situación con un mensaje similar a este:

"Casillero ya descubierto, aquí hay agua/impacto, pierde su turno..."

Las jugadas se realizan por turnos, primero el usuario, luego el sistema y así sucesivamente.

El sistema no está dotado de inteligencia, e propondrá casilleros (válidos) al azar, con la salvedad que el sistema no repita más de dos veces una misma jugada, esto debe ser asegurado por la implementación.

Cuando el usuario impacta por completo o hunde un barco (por ejemplo, acertando los 5 casilleros correspondientes al portaaviones), deberá aumentarse el contador del tipo de barco destruido y se advertirá esto con un mensaje por pantalla.

El primer jugador que logre los 29 impactos, ganará el juego, lo cual debe indicarse por pantalla.

El ofrecerá al usuario jugar otra partida antes de terminar.

Importante

- El programa se puede realizar en **grupo de hasta 2(dos) integrantes**
- El programa entregado **deberá compilar y ejecutar el total de las funcionalidades requeridas**
- Deberá presentarse la **documentación** del código explicando:
 - el o los algoritmos más importantes desarrollados o funciones y estructuras de datos destacadas y su propósito, expresándolos en lenguaje natural o algún esquema descriptivo
 - indicaciones para la compilación y ejecución del programa (preferentemente mediante **make**)
 - que funciones o algoritmos responden a cada uno de los requerimientos expresados
 - expresar **mejoras futuras**, principalmente cómo haría para dotar de un nivel de inteligencia superior al sistema cuando propone jugadas
 - observe el ejemplo del Buscaminas subido a la web para ver un ejemplo de documentación
 - **la documentación debe entregarse junto con el programa vía mail**
- El código fuente debe ofrecer **comentarios adecuados** que expliquen las funcionalidades provistas por cada función y detalles de construcciones o bucles que puedan resultar de difícil comprensión, y ceñirse, al menos en parte, a **reglas de coding-style**
- Antes de entregar el sistema realice las pruebas suficientes para asegurarse que se cumplen los requerimientos y que el usuario puede ganar y perder partidas.
- El trabajo se deberá presentar en la **mesa de examen final correspondiente a la materia**.
- Podrán realizarse **consultas** sobre diseño e implementación durante las clases restantes, días de consulta y mesas de examen.