

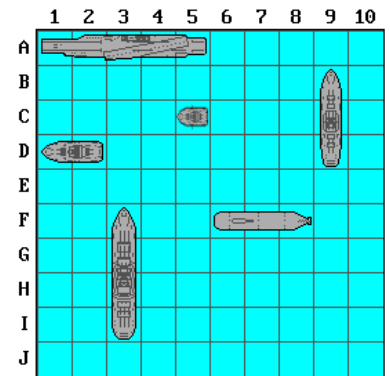
Proyecto Programado #3

Batalla Naval

Batalla Naval es un juego de mesa cuyo objetivo es encontrar los barcos del oponente y hundir toda su flota mediante disparos o lanzamientos al campo de batalla.

Este juego tiene sus orígenes en los años 1900 y se jugaba con lápiz y papel, dibujando una cuadrícula para simular el campo de batalla y se ubicaban los barcos sobre ella.

Se jugaba principalmente entre los soldados de la primera guerra mundial, sin embargo, fue hasta el año 1967 que se publicó tal y como lo conocemos hasta el día de hoy.



¿Qué se busca con este proyecto?

El objetivo general de este proyecto es facilitar un acercamiento con el juego de batalla naval desde la perspectiva de la programación, de manera que las partes del juego y las reglas que lo enmarcan sean diseccionadas minuciosamente para desarrollar un diseño que sea programado mediante el lenguaje Python.

1. Practicar las habilidades del aplicaciones de software
2. Ejercitar la toma de decisiones sobre el dominio del problema y de la solución
3. Aplicar los conceptos de programación y manejo de matrices.
4. Manejo de componentes para el uso de interfaz gráfica
5. Utilización de la programación orientada a objetos para la implementación de la aplicación

Descripción del juego

El juego está diseñado para jugarse entre 2 jugadores. Se juega sobre dos tableros, cada jugador tiene un tablero que consiste de dos cuadrículas de 10x10 cada una. En una de ellas se colocaran los barcos. En la otra cuadrícula, se irán colocando “disparos” con el objetivo de ubicar la flota del adversario y hundirla completamente. Gana el primero en lograrlo.

Al iniciar el juego, cada jugador posiciona sus barcos en el primer tablero, de forma secreta, de tal forma que quede invisible al oponente.

Cada quien ocupa, según sus preferencias, una misma cantidad de casillas, horizontal y/o verticalmente, las que representan sus embarcaciones. Ambos participantes deben ubicar igual cantidad de embarcaciones. Para efectos del juego, **cuatro casillas consecutivas conforman un portaaviones; tres un buque, y una casilla aislada, una lancha.**

Adicionalmente, cada jugador tendrá derecho a las siguientes cantidades: **dos portaaviones, tres buques y cuatro lanchas.** No se podrán colocar barcos pegados entre sí en la misma dirección.

Una vez que todas las embarcaciones han sido posicionadas, se inicia una serie de turnos. En cada turno, el jugador “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 un barco contrario, se deberá indicar visualmente la afectación de la nave contraria, si todavía quedan partes del barco (casillas) sin dañar, o “Hundido” si con ese disparo la nave ha quedado totalmente destruida (esto es, si la acertada es la última de las casillas que conforman la nave que quedaba por acertar). Si la posición indicada no corresponde a una parte de barco alguno, se mostrará de alguna manera el disparo fallido.

Así se continuará jugando hasta que un jugador destruya en su totalidad las embarcaciones del oponente.

Proyecto a desarrollar

Este proyecto requiere de un conocimiento detallado del juego de batalla naval, por lo que se recomienda que antes de iniciar el trabajo debe comprender por completo el funcionamiento del juego, ya que lo que se desea es reproducir el comportamiento del mismo.

Asegúrese de conocer a la perfección:

1. El modos de juego
2. Diseño del juego
3. Dimensiones
4. Desarrollo del juego
5. Finalizaciones del juego
 - a. Gane
 - b. Abandono

El juego debe disponer de las siguientes funcionalidades:

Inicio del Juego:

Debe desarrollar la funcionalidad para generar un nuevo juego, donde el usuario podrá personalizar el juego por medio de las siguientes opciones:

- a) Seleccionar el tipo de juego:
 - i. Normal: los jugadores resuelven el juego durante el tiempo que deseen
 - ii. Cantidad de Disparos: el usuario elige la cantidad de disparos que realizará cada jugador. Cuando ambos jugadores realicen todos los disparos. Se analiza cuál causó más daño y ese será el ganador.
- b) Seleccionar el modo de juego:
 - i. Multi-jugador: el juego se resolverá entre dos jugadores que utilizan el juego de manera intercalada. Para el cambio de turno se debe presionar un botón de continuar.
 - ii. Vs Computadora: un jugador compite contra la Computadora. Se debe programar un algoritmo inteligente que permita a la computadora jugar. Debe elegir el mejor tiro, por ejemplo, selecciona aleatoriamente un lugar de disparo, si acierta debe validar si la nave fue eliminada completamente, si no es así, el siguiente disparo intentará destruirlo por completo hasta que lo logre. El cambio de turno se realiza automáticamente.
- c) Selección de Nombres: si el modo de juego es Multi-jugador se habilitan dos espacios para indicar los nombres de los jugadores, si es modo Vs Computadora sólo se solicita un nombre.
- d) Seleccionar ambiente de juego: se debe disponer de cuatro ambientes de juego: Pacífico Central, Región Polar, Piratas del Caribe y Vikingos. El tablero y las embarcaciones deben ir acorde a cada ambiente de juego. Debe mostrar imágenes en miniatura asociadas cada ambiente de juego para la selección de ambiente.
- e) Ubicación de barcos: cada usuario podrá ubicar cada una de las embarcaciones dentro de su área de juego. Por cada nave puede elegir una ubicación válida y una orientación válida (arriba-abajo, abajo-arriba, izquierda-derecha, derecha-izquierda) dentro de su cuadrante de juego. Además debe disponer de una opción para ubicar todos sus barcos aleatoriamente.
- f) Iniciar: se le muestra al primer jugador el área de juego. Se debe elegir aleatoriamente si inicia el jugador uno o el jugador dos.

Desarrollo de juego:

Durante el desarrollo del juego la ventana debe disponer de lo siguiente:

- a) Ambiente de juego, debe mostrar el mapa en el cuadrante y las embarcaciones acorde al ambiente.
- b) Mostrar el cuadrante de juego para cada jugador. A cada jugador muestra su cuadrante con el estado de sus embarcaciones y en el cuadrante enemigo donde ha disparado y donde ha causado daño.
- c) Debe mostrar el estado de las embarcaciones propias: indicando el estado (porcentaje de daño según su tamaño y disparos recibidos).
- d) Debe mostrar el estado de las embarcaciones enemigas: si están en combate o si fueron destruidas por completo.
- e) Debe ir intercambiando entre ambos jugadores el área de juego y los disparos.
- f) Disparar: según el turno cada jugador elige la celda de ataque (un único disparo por turno) y debe simular el disparo:
 - i. Si no daña ninguna embarcación debe observarse un disparo únicamente.
 - ii. Si causa daño pero la embarcación no ha sido destruida totalmente, muestra el disparo y una explosión.
 - iii. Si causa daño y la embarcación es destruida totalmente, muestra el disparo, una explosión mayor y en el tablero de embarcaciones enemigas se muestra la inhabilitación de la embarcación según el tipo destruido.
- g) Cantidad de disparos: por cada jugador debe indicar la cantidad de disparos que ha realizado.
- h) Botón de abandono: en cada turno el jugador puede abandonar, y se debe acreditar el gane al jugador oponente.
- i) Indicar gane: después de que ambos jugadores finalicen una ronda de disparos debe validar si algún jugador ganó o si hay empate, si es el caso indicarlo y mostrar la ventana de estadísticas de juego (ver abajo).

Estadísticas por juego:

Una vez que finalice el juego se debe indicar lo siguiente:

- a) Nombre del ganador (en caso de empate indicarlo y mostrar el nombre de ambos jugadores).
- b) Tiempo invertido.
- c) Cantidad de disparos.

Características opcionales

La siguiente característica no es obligatoria, pero se asignarán puntos extras en caso de que se desarrollen. Sólo se otorgarán puntos extra si las funcionalidades descritas anteriormente están completas.

- Repetición de partida. Permitir al usuario la opción para ver una repetición de la partida, mostrando una jugada por segundo o similar. 10 pts

Aspectos técnicos

El proyecto deberá estar escrito en el lenguaje de programación Python y se deben desarrollar las funcionalidades de usuario por medio de interfaz gráfica.

En el desarrollo del juego sólo se permiten acciones válidas (investigar según se indica arriba).

Se debe poder navegar siempre a la ventana principal del juego.

Y considerar lo siguiente:

- Se deben manejar mensajes claros al usuario
- Realizar validaciones de captura de campos
- Toda función built-in de Python que deseen utilizar debe ser validada con el profesor
- En el desarrollo del programa se deberá utilizar clases (por cada archivo .py puede existir una única clase) e iteración.

Documentación

El código fuente debe tener documentación interna, con comentarios precisos y bien ubicados. Cada función debe tener descripción, entradas, salidas y restricciones.

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas, restricciones y **objetivo**.

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: **instrucciones de compilación, ejecución y uso**.
3. Pruebas de funcionalidad: incluir *screenshots*.
4. Descripción del problema.
5. Diseño del programa: decisiones de desarrollo, algoritmos usados.
6. Librerías usadas: creación de archivos, etc.
7. Análisis de resultados: objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).
8. Bitácora del trabajo realizado
9. Conclusión (es)

Evaluación

La evaluación se va a centrar en dos elementos: programación y documentación.

El proyecto programado tiene un valor de **25%** de la nota final, en el rubro de Proyectos.

Desglose de la evaluación del proyecto programado:

1. Documentación interna 4 ptos.
2. Documentación externa 6 ptos.
3. Funcionalidad 75 ptos (ver detalle en Software a Desarrollar)
4. Revisión del proyecto (según completitud del proyecto) 10 ptos.
5. Hora de Entrega 5 ptos.

Forma de trabajo

El trabajo se debe realizar en parejas.

Aspectos administrativos

Debe crear un archivo **.zip** ("PP3.zip") que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentacion** y **programa**, en la primera deberá incluir los documentos de *word* (no pdf) solicitados, y en la segunda los archivos y/o carpetas necesarias para la implementación de esta tarea. El archivo **info.txt** debe contener la siguiente información (cualidades):

- a. Nombre del curso
- b. Número de semestre y año lectivo
- c. Nombre del Estudiante
- d. Número de carnet del estudiante
- e. Número del proyecto programado
- f. Fecha de entrega
- g. Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada):
[Deplorable | Regular | Buena | MuyBuena | Excelente | Superior]

Entrega

Deberá subir el archivo antes mencionado al TEC Digital en el curso de TALLER DE PROGRAMACIÓN GR 61, en la asignación llamada "Proyecto III" debajo del rubro de "Proyectos". En la evaluación del Proyecto el rubro de "Hora de Entrega" valdrá 5 puntos de la nota total del proyecto, según la siguiente escala:

- a. Si se entrega antes de las 11:55:55 PM del martes 12 de junio de 2018, 5 puntos.
- b. Si se entrega antes de las 11:55:55 AM del miércoles 13 de junio de 2018, 2.5 puntos.
- c. Si se entrega antes de las 11:55:55 PM del miércoles 13 de junio de 2018, 0 puntos.

Después de este punto, **NO SE ACEPTARÁN** más trabajos.

Los archivos fuentes pueden ser revisados en el sistema de Control de Plagio del TEC Digital. **Todo el código de cada proyecto debe ser 100% original y en caso de plagio se asignará nota cero.**

Bibliografía

[https://es.wikipedia.org/wiki/Batalla_naval_\(juego\)](https://es.wikipedia.org/wiki/Batalla_naval_(juego))

<http://www.juegos.com/juego/batalla-naval>