



501125

Introducción a la Programación Tarea Semestral

JAVIER VIDAL VALENZUELA

25 de mayo de 2023

Introducción

El propósito de esta tarea es familiarizarse con el proceso de análisis, diseño e implementación de la solución a un problema de programación de mediana a alta complejidad usando lenguaje Python. También se espera que a través de la experiencia en este trabajo se potencien las habilidades de trabajo en equipo.

El proyecto **debe ser realizado en grupos de 4 alumnos/as (excepcionalmente se permitirán grupos de 3 estudiantes)**, quedando absolutamente prohibido el trabajo individual o en una configuración distinta de equipos.

Este enunciado se entregará a través de la plataforma CANVAS y en el caso de requerir aclaración de alguno de los aspectos expuestos en el documento, se recomienda que la comunicación se haga electrónicamente, de preferencia usando la misma plataforma, para hacer extensivas las explicaciones al resto del curso.

Batalla Naval

La batalla naval es un juego tradicional de estrategia y algo de suerte, que involucra a dos participantes.

Cada jugador maneja un tablero que representa el océano y un tablero de tiro, del mismo tamaño que el anterior; cada uno dividido en casillas, enumerando con letras A, B, C, etc. y/o con números 1, 2, 3, etc. las filas y/o las columnas.

Cada tablero representa una zona diferente del mar abierto: la propia y la contraria. En el primer tablero, el jugador coloca sus barcos y registra los «tiros» del oponente; en el otro, se registran los tiros propios contra el otro jugador, diferenciando los impactos y los que dan al agua.

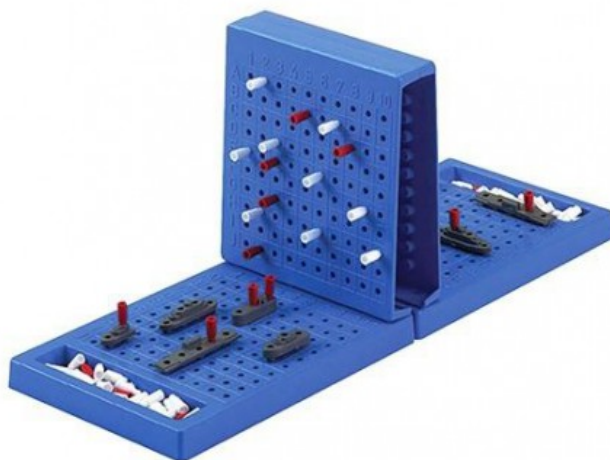
Para comenzar el juego, cada jugador posiciona sus barcos en el primer tablero, de forma secreta, invisible al oponente. Cada quien ocupa, según sus preferencias, una misma cantidad de casillas, horizontal y/o verticalmente, las que representan sus naves. Ambos participantes deben ubicar igual número de naves, por lo que es habitual, antes de comenzar, estipular de común acuerdo la cantidad y el tamaño de las naves que se posicionarán en el tablero. Así, por ejemplo, cinco casillas consecutivas conforman un portaaviones; cuatro, un buque; tres, un submarino; dos, un crucero; y una casilla aislada, una lancha. Los participantes podrían convenir, por ejemplo, colocar, cada uno, dos portaaviones, tres buques y cinco lanchas. No se podrán colocar barcos pegados entre sí.

Una vez que ambos jugadores han posicionado todas las naves, se inicia una serie de rondas.

En cada ronda, cada jugador en su turno «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, el oponente cantará «¡Impacto!» 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). El jugador que ha tocado un barco en su anterior jugada, volverá a disparar hasta que falle. Si la posición indicada no corresponde a una parte de barco alguno, cantará «¡Agua!».

Cada jugador referenciará en ese 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.

El juego termina con un ganador, que será quien destruya primero todas las naves de su oponente.



Para la tarea se pide implementar el juego de la batalla naval con las siguientes restricciones:

- El tamaño del tablero será de $N \times N$, con $N \geq 10$.
- Las filas y columnas del tablero se referenciarán usando números enteros mayores o iguales a 0.
- El oponente será el computador, que generará su tablero y sus jugadas siguiendo las normas del juego y en forma aleatoria.
- La cantidad de naves de cada jugador debe ser consultada al usuario del juego (debe ser un número entero mayor que 2 y menor o igual a N)
- Todas las naves serán buques de tamaño 3 casillas.

Entradas:

La entrada a este programa consistirá en varias líneas. La primera corresponde a un entero positivo N ($10 \leq N \leq 1000$) el cual especifica el tamaño del tablero y la cantidad máxima de naves de cada jugador. Luego, vendrá N pares de números y una letra ('H' o 'V') que especificará las coordenadas del punto central y la orientación (horizontal o vertical) de cada nave. Con esta información se puede generar el registro del resto de la nave (proa y popa).

En seguida, se debe especificar quién parte jugando, indicando “Humano” o “Máquina”.

A continuación, sucesivamente con el oponente (el computador) se deben ir ingresando (generando) pares de números que representan las coordenadas de los disparos. El programa debe indicar si se trata de un impacto o agua y debe generar los registros correspondientes.

Cada una de estas entradas deben ser validadas o en el caso de la información de las naves y coordenadas de los disparos se deben ajustar al tamaño del tablero. En el caso que los valores no estén dentro de los rangos establecidos, se debe indicar el error y permitir un nuevo ingreso de datos.

Salidas:

Habr  s lo una salida del programa consistente en uno de los siguientes textos “Gana el humano” o “Gana la m quina”.

Instrucciones

Este trabajo debe ser realizado por equipos que se deben registrar en CANVAS, cuyos integrantes deben designar un/a **coordinador/a** o **l der**, quien ser  el/la que se comunique con el/la profesor/a en caso de requerirlo.

La conformaci n de los equipos debe registrarse antes del **viernes 2 de junio de 2023**.

La entrega del trabajo tendr  un **plazo m ximo que vence el s bado 1 de julio de 2023**, para lo cual cada equipo debe preparar lo siguiente:

1. un archivo denominado **INFORME.PDF** que contenga:
 - una **portada** en que figuren, como m nimo, los nombres de los/as integrantes del equipo.
 - En su interior, el informe debe establecer **qu  funciones fueron asignadas a cada miembro del grupo de trabajo**, **cu ntas reuniones realizaron** y **cu l fue el objetivo de cada una**, indicando adem s **qui nes participaron en cada reuni n**, es decir, la **organizaci n del equipo**,
 - se debe adem s describir el **desarrollo** indicando **la forma en que se realiz  el trabajo** (**an lisis del problema** (descripci n detallada de las especificaciones no indicadas en el enunciado del problema), **estrategias de soluci n** (algoritmos, funciones, estructuras de datos, variables usadas, etc. que implementan la soluci n al problema) e incluya algunas **pruebas** realizadas con datos.
 - Finalmente, el informe debe incluir tambi n algunas **conclusiones generales** del trabajo.

No incluya c digo en el informe.

2. un archivo denominado **readme.txt** que contenga **los nombres de los integrantes del equipo y las instrucciones**, tanto para ejecutar su programa. Las indicaciones en este archivo se seguir n estrictamente, por lo tanto, es su responsabilidad ser claro en el orden y especificidad de las instrucciones.
3. uno o m s archivos ***.py** o ***.ipynb** **con el c digo en lenguaje Python**. Los programas deben incluir (en l neas de comentarios) los nombres completos de los integrantes del grupo adem s de comentarios que permitan identificar las partes principales de los programas.

Todos los archivos mencionados deben comprimirse en un paquete ZIP o RAR con el nombre identificatorio de su equipo, por ejemplo “Equipo.001.ZIP”, el cual debe ser enviado a trav s de la plataforma CANVAS.

No se aceptar n tareas incompletas, esto es, la tarea consiste en el programa funcionando y el informe. Si falta alguna de estas secciones el resultado ser  una nota reprobatoria.

- Asegurarse de escribir la informaci n de identificaci n del grupo en todos los archivos.
- Eliminar despliegues innecesarios en el(los) programa(s).
- No ser n recibidas partes o la totalidad de la tarea en forma impresa.
- No env e archivos ejecutables (*.EXE), la plataforma y algunos clientes de correo electr nico no lo permiten.
- Cuando env e su trabajo, el profesor confirmar  si su tarea ha sido recibida sin problemas, si esto no ocurre, consulte al profesor.
- Si despu s de enviar la tarea descubre un error y el grupo desea enviar una copia revisada, hacer otro paquete ZIP o RAR, y volver a depositar (nos quedaremos con la  ltima copia enviada).

Evaluación

El código será compilado, ejecutado y probado en una máquina usando WINDOWS y JUPYTER NOTEBOOK, por lo tanto, si algún grupo requiere la instalación de alguna biblioteca especial o aplicación complementaria, o bien, si es necesario compilar o correr la aplicación en un sistema operativo diferente, debe informarlo al profesor usando el archivo `readme.txt`.

El puntaje máximo de la tarea es 60 puntos distribuidos de la siguiente forma:

- 1 Entrega oportuna.
- 3 Cumple con especificaciones de entrega del trabajo incluyendo la estructura definida para el informe.
- 3 Describe adecuadamente la organización del equipo.
- 10 Realiza un análisis del enunciado y datos con el fin de diseñar una estrategia de solución al problema.
- 10 Describe las estrategias de solución al problema.
- 10 Describe los algoritmos, las funciones y las estructuras de datos utilizadas.
- 3 Expone conclusiones adecuadas al desarrollo del trabajo.
- 20 Correctitud del código cuando es ejecutado. Se realizará una prueba exhaustiva de todas las funciones solicitadas, incluyendo ingreso de datos fuera de rango, para los cuales el programa debe responder correctamente.

El informe debe estar escrito correctamente, bien diagramado y sin problemas de ortografía y redacción (se descontará 2 décimas de su nota por cada falta de ortografía en el informe).

La información contenida en el informe será evaluada de acuerdo a si esta aporta a la comprensión de la solución al problema. La solución debería ser clara y directa, por tanto, los comentarios deberían ser informativos y no extensos.

Recuerde que no puede presentar sólo el informe o sólo el código. Se deben entregar las dos cosas.

Compra y copia de tareas

Se ha detectado una red de compra y venta de las tareas semestrales, lo cual constituye una falta de ética grave no aceptada en alumnos de la Universidad de Concepción.

Es por ello que habrá una exhaustiva verificación de la autenticidad de las soluciones planteadas por los alumnos del curso. Esto significa que los alumnos integrantes de los equipos que no sean capaces de explicar los detalles técnicos de sus soluciones planteadas, tanto a nivel de algoritmos como a nivel de código, serán evaluados con nota mínima obteniendo la reprobación de la asignatura con condición NCR (No Cumple Requisitos).

Recomendación

La tarea puede ser desarrollada con cualquier intérprete usando el sistema operativo que estime conveniente, sin embargo, antes de entregar sus resultados se recomienda asegurar que estos compilen y ejecuten en JUPYTER NOTEBOOK sobre WINDOWS.

Sobre el Informe

Para confeccionar el informe de la tarea se recomienda seguir las pautas para la confección de un informe de memoria de título para las diferentes carreras en la Facultad de Ingeniería. No obstante lo anterior, a continuación se entregan algunas indicaciones para sus principales secciones:

-
- **Portada** El informe debe contar con una portada consistente en una página al inicio del informe con la siguiente estructura:
 - en la parte superior derecha (encabezado) debe contener el logo de la Universidad de Concepción con el texto "Facultad de Ingeniería" en la parte inferior de este el texto "Universidad de Concepción".
 - en la parte superior izquierda (encabezado) debe contener el logo de la Facultad de Ingeniería.
 - en la parte central debe contener el texto "Introducción a la Programación" en la parte inferior de este el texto "Tarea Semestral"
 - en la parte inferior derecha debe contener el nombre de todos los integrantes del equipo.
 - en la parte central inferior (pie de página) debe contener la fecha con el formato "Junio de 2023"
 - **Introducción** Una o dos páginas que resuman los aspectos más importantes del trabajo.
 - **Organización** Una o dos páginas en las que se identifique las funciones que asumieron como responsables los diferentes integrantes del equipo. Se solicita indicar no sólo el nombre de la función sino una breve descripción de sus principales actividades o responsabilidades. Incluya acá la lista de reuniones sostenidas, medio usado para la reunión, asistentes, objetivo de la reunión, resultado de la reunión.
 - **Desarrollo** Esta sección consta de varias subsecciones como por ejemplo:
 - Análisis, cuando se enfrenta al desafío de resolver un problema de mediana o alta complejidad con programación, antes de comenzar con el programa e incluso con la definición de los algoritmos, es necesario analizar cuáles son los datos, qué estrategias, qué modelos, qué procedimientos, qué fórmulas, etc. se utilizarán para pasar a la solución. Esto debe ser descrito en la sección de análisis del problema y estrategias de solución. En esta sección pueden aparecer incluso diferentes estrategias de solución las que posteriormente se pueden ir descartando para elegir la mejor o la factible.
 - Estructuras de datos, permite identificar que variables, arreglos, listas, etc. se requerirán para resolver el problema y qué información se almacenará en cada una de estas estructuras.
 - Algoritmos, permite describir los grandes bloques de funciones que se ejecutarán para resolver el problema. No es necesario describir algoritmos con niveles detallados de los procedimientos ya que esto no resultaría útil para problemas de mediana o alta complejidad.
 - **Resultados** En esta sección se puede documentar la ejecución del programa con algunos datos de entrada específicos. La descripción puede venir acompañada de algunos pantallazos correspondientes a la ejecución del programa.
 - **Conclusiones** Todo trabajo con las características del exigido en este curso requiere que una vez finalizado los ejecutores reflexionen y describan los aspectos más relevantes de su realización, respuestas preguntas como ¿cumplimos el objetivo del trabajo?, ¿qué aprendimos?, ¿qué faltó aprender o resultó más complejo de realizar?, ¿qué decisiones tomamos que facilitaron la realización del trabajo?, etc. puede responderse en esta sección. No se extienda por más de una página en esto.