PROYECTO III

AirWar

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores
Algoritmos y Estructuras de Datos I (CE 1103)
II Semestre 2024

Objetivos

General

- Implementar un juego que permita la aplicación del Paradigma Orientado a Objetos mediante la utilización de estructuras de datos generales y algoritmos de búsqueda y ordenamiento.

Específicos

- Implementar una solución que permita resolver el problema descrito en esta especificación utilizando Programación Orientada a Objetos en C# y haciendo uso de estructuras de datos generales.
- Elaborar la documentación correspondiente a la solución implementada para la evidencia del trabajo desarrollado utilizando estándares de documentación técnica y herramientas de gestión de proyectos.

Lista de requerimientos

AirWar es un juego de guerra aérea. Cuando el jugador inicia AirWar, se muestra un mapa del mundo. Se generarán aleatoriamente aeropuertos y portaaviones en distintas posiciones del mapa. En la parte inferior de la ventana del juego, hay una batería antiaérea que el jugador controla para tratar de destruir los aviones en ruta.

A continuación, se detalla cada uno de los requerimientos

ID	DESCRIPCIÓN	PUNTOS
000	El juego se desarrollará en C# con WPF, MAUI o Unity.	Binario
001	El objetivo del juego es destruir la mayor cantidad de aviones en un periodo de tiempo definido por el estudiante.	10
002	Generación aleatoria de aeropuertos, portaaviones y rutas. Se modela con un grafo utilizando listas de adyacencia.	10

003	Batería antiaérea que se mueve en velocidad constante entre izquierda y derecha de la pantalla. El jugador presiona click para disparar balas con trayectoria recta y de velocidad variable según el tiempo que se presionó el click antes de lanzar la bala.	15
004	Entre aeropuertos y portaaviones, se generarán rutas aleatorias con distintos pesos, considerando que: • Una ruta tendrá un peso general dado por la distancia entre los puntos que conecta. • El peso de una ruta considerará también el destino que conecta. Es más caro aterrizar en un portaaviones que en un aeropuerto y es más caro seguir una ruta interoceánica (que atraviesa el océano) que una continental.	10
005	Cuando un avión va a despegar, decide a qué destino quiere ir aleatoriamente y calcula la mejor ruta a ese destino, considerando los pesos anteriormente dados. Cuando un avión aterriza, dura una cantidad aleatoria de segundos antes de retomar una nueva ruta. Durante el tiempo de espera, el avión recarga una cantidad aleatoria de combustible.	10
006	No necesariamente todos los aeropuertos tienen suficiente combustible para todos los aviones, por lo que lo racionan, de alguna forma determinada por el estudiante. Un avión puede caerse si se le acaba el combustible antes de aterrizar.	5
007	Cada cierto tiempo los aeropuertos deben construir nuevos aviones. Por regulaciones internacionales, cada aeropuerto no podrá generar más aviones que la que permita sus hangares; por lo tanto, uno de los atributos de los aeropuertos es la cantidad de aviones que soporta su hangar. Los aviones deben tener un ID generado aleatoriamente utilizando GUIDs.	10
008	Suponga que los aviones son realmente drones autónomos (se manejan solos), pero que contienen cuatro módulos de AI independientes que trabajan juntos para volar el avión. Los módulos de AI se llaman: 1. Pilot: maneja el avión activamente. 2. Copilot: se encuentra en standby y tomará control activo del avión únicamente si algo sucede con el módulo Pilot. 3. Manteinance: identifica e intenta corregir cualquier problema en las piezas físicas del avión. 4. Space awarness: es el módulo encargado de monitorear los sensores y radares, para darle insumos al módulo Pilot. Y de cada uno de ellos, interesa conocer su 1. ID: es un identificador que consta de tres letras (de la A a la Z) generadas aleatoriamente a la hora de crear el avión. 2. Rol (piloto, copiloto, asistente) 3. Horas de vuelo	10

	El sistema debe permitir obtener la lista de aviones derribados y ordenarlos por su ID usando Merge Sort.	
009	Para cada avión derribado, es posible obtener su tripulación (módulos Pilot, Copilot, Manteinance y Space Awarness) y ordenarlo por ID, rol o horas de vuelo, utilizando Selection Sort.	10
010	En pantalla se debe mostrar todos los datos relevantes del juego. Por ejemplo, se deben mostrar los caminos calculados por los aviones, los pesos de cada ruta, los atributos de los aviones, entre otros.	10
	Total	100

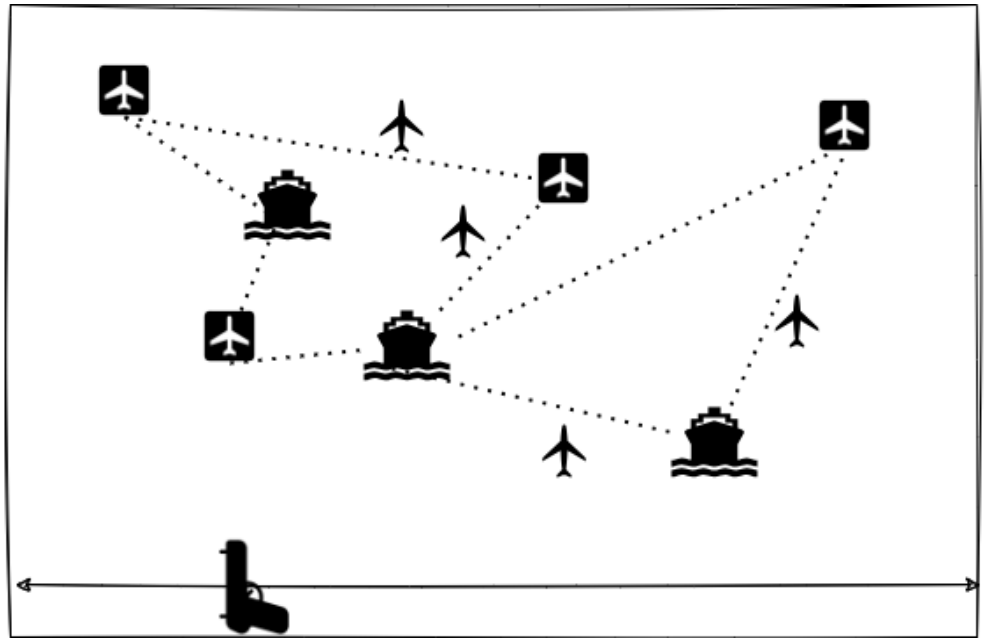

Figura 1. Ejemplo de interfaz gráfica para el juego

Opcional

Podrá optar por un 20% extra si crea una maqueta para mostrar la batería antiaerea. Se sugiere utilizar un Arduino y un servomotor para este propósito. La maqueta debe

- Mover el cañón de izquierda a derecha en sincronía con el cañón que aparece en la GUI
- Encender un LED cada vez que haya un disparo
- Permitir disparar desde la maqueta, utilizando un botón

Queda a criterio del profesor cómo se califica esta sección.

Documentación

Este proyecto no requiere de documentación externa, aparte del documento de acreditación que se explica en la siguiente sección.

Documento de acreditación

Objetivo general

- Diseñar soluciones creativas para problemas de ingeniería complejos, diseñando sistemas, componentes o procesos para satisfacer las necesidades identificadas con la consideración adecuada para la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como las consideraciones de recursos, culturales, sociales y ambientales según sea necesario.

Objetivos específicos

- Identificar las necesidades y los requerimientos de un problema complejo de ingeniería considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.
- Valorar alternativas de solución para un problema complejo de ingeniería que cumpla con necesidades específicas, considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.
- Diseñar de forma creativa, la alternativa seleccionada que cumpla con las necesidades específicas para resolver el problema complejo de ingeniería, considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.
- Valida el diseño final de acuerdo con los requerimientos, la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.

Atributos de acreditación a evaluar en este proyecto

- Diseño en nivel inicial.

Descripción del entregable

Cada grupo debe elaborar un documento PDF que tenga la siguiente estructura:

1. Portada.

2. Tabla de contenidos.
3. Introducción.
4. Diseño
    a. Listado de requerimientos descritos con el formato de historias de usuario.
    b. Seleccione cinco problemas que tengan más de una posible solución, y para cada uno de ellos haga lo siguiente
        i. Proponga y explique dos alternativas de solución
        ii. Para cada alternativa, deje claro cuáles son las ventajas y cuáles las desventajas
        iii. Seleccione una de las dos alternativas y explique por qué la seleccionó
    c. Elabore un diagrama de clases usando UML
    d. Elabore un diagrama de arquitectura
    e. Describa, mediante un checklist, las historias de usuario que fueron implementadas y las que quedaron pendientes.

Aspectos operativos

- El trabajo se realizará en **parejas**.
- El uso de Git y Github es obligatorio
- La fecha de entrega será según lo especificado en el TEC Digital. Se entrega en el TEC digital, un archivo PDF con el documento de acreditación. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la cita revisión oficial.
- El proyecto tiene un valor de 15% de la nota del curso
- Los proyectos que no cumplan con los siguientes requisitos no serán revisados:
    - Toda la solución debe estar integrada
    - La interfaz de usuario debe estar implementada e integrada
- La funcionalidad tendrá un valor total de 80%, y el documento de acreditación valdrá 20%. De estas notas se calculará la *Nota Final del Proyecto*.
- Aun cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
    - **Si no se entrega el documento de acreditación en formato PDF, automáticamente se obtiene una nota de 0.**
    - Si no se utiliza un manejador de código se obtiene una nota de 0.
    - Si la documentación no se entrega en la fecha indicada se obtiene una nota de 0.
    - El código debe desarrollarse en C#, si no, se obtendrá una nota de 0.
- La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto.
- Cada estudiante tendrá 5 minutos para exponer su trabajo al profesor y defenderlo, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo que se recomienda tener todo listo antes de entrar a la defensa.

- Cada grupo es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión a el profesor para coordinar el préstamo de estos.
- Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador de la carrera.