

Proyecto #3 - AirWar

Tecnológico de Costa Rica
Área Académica Ingeniería en Computadores
CE-1103 Algoritmos y Estructuras de Datos I
I Semestre 2023
Valor 15%



Objetivo 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.

Objetivos específicos

- Diseñar una solución que permita resolver el problema descrito en esta especificación aplicando patrones de diseño y utilizando el estándar UML.
- Implementar una solución que permita resolver el problema descrito en esta especificación utilizando Programación Orientada a Objetos en Java 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.

Descripción del problema

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. De igual forman entre aeropuertos y portaaviones, se generarán rutas aleatorias con distintos pesos, por ejemplo:

1. Una ruta tendrá un peso general dado por la distancia entre los puntos que conecta.
2. 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 una continental
3. El peso de la ruta considerará el peligro. Cada vez que el usuario destruye un avión o un avión se estrella por falta de combustible, dicha ruta aumentará en peligro. Conforme el tiempo pasa y el usuario no destruye aviones ni se estrellan aviones en dicha ruta, el peligro se ajustará nuevamente.

Cuando un avión va a despegar, decide a dónde quiere ir aleatoriamente y calcula la ruta óptima, 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. No necesariamente todos los aeropuertos tienen suficiente combustible para todos, 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.

Los ingenieros constantemente están haciendo mejoras a los aviones. Cada avión tiene ciertos atributos. Por ejemplo:

1. Velocidad
2. Eficiencia (consumo de combustible)
3. Fortaleza

Los ingenieros de aviones envían cada cierta cantidad de tiempo órdenes a cada aeropuerto para construir nuevos aviones. Por regulaciones internacionales, cada aeropuerto no podrá generar más aviones que la que permita sus hangares. El sistema debe tener una lista de opciones para crear aviones, donde estos tipos de aviones tienen valores predefinidos para velocidad, eficiencia y fortaleza. Cada tipo de estos aviones tienen un nombre.

Para crear un avión el usuario puede buscar el tipo por nombre, o bien, puede pedir que los tipos se ordenen por Velocidad (Insertion Sort) o Eficiencia (Shell Sort).

Todos los datos relevantes del juego se deben mostrar. Por ejemplo, se deben mostrar los caminos calculados por los aviones, los pesos de cada ruta, los atributos de los aviones, entre otros.

AirWar se compone de dos aplicaciones: una aplicación que es el juego en sí y una aplicación que sirve como controlador aéreo. El controlador aéreo puede ver en todo momento el estado del juego y las estadísticas. Así mismo puede cancelar rutas temporalmente. La comunicación entre ambas aplicaciones debe ser mediante sockets.

Controller

Implementado en Arduino. Debe permitir:

1. Seleccionar un avión para intentar destruirlo.
2. Mostrar 5 leds que se van encendiendo de manera secuencial, uno por uno, de extremo a extremo.
3. El jugador debe intentar apretar un botón en el *Controller* de manera que los leds se detengan en el led del centro.
4. Si el jugador logra seleccionar el led del medio debe emitir un sonido cuando confirme que fue eliminado el avión seleccionado.

Requerimientos no funcionales

1. Cabe resaltar que es de manera obligatoria realizar la maqueta del control para interactuar con la aplicación. No se permite presentar el circuito únicamente en la protoboard.
2. Bibliotecas para comunicación serial con Arduino: jSerialComm o RXTX.

Documentación requerida

1. Internamente, el código se debe documentar utilizando Javadoc. Se debe generar el HTML.
2. La documentación externa se hará en un documento que incluya lo siguiente (deberá entregarse un PDF):
 - a. Breve descripción del problema.
 - b. Diagrama de clases.
 - c. Descripción de las estructuras de datos desarrolladas.
 - d. Problemas encontrados en forma de bugs de *github*: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
3. **Planificación y administración del proyecto:** se utilizará Azure DevOps para la administración del proyecto. Debe incluir:
 - a. Lista de features e historias de usuario identificados de la especificación.
 - b. Plan de iteraciones que agrupen cada bloque de historias de usuario de forma que se vea un desarrollo incremental
 - c. Descomposición de cada user story en tareas.
 - d. Asignación de tareas entre las personas integrantes del grupo.

Aspectos operativos y evaluación:

1. **Fecha de entrega: De acuerdo con el cronograma del curso y lo establecido en el TEC Digital**
2. El proyecto tiene un valor de 15% de la nota del curso.
3. El trabajo es **en grupos de tres personas**.
4. Deben entregar en el TEC Digital un documento con el link del repositorio de GitHub, Azure DevOps y el PDF de la documentación. Para ambas herramientas deben dar acceso al correo del profesor.
5. Es obligatorio utilizar un Git y GitHUB para el control de versiones del código fuente y evidenciar el uso de Commits frecuentes.
6. Es obligatorio integrar toda la solución.
7. El código tendrá un valor total de 60%, la implementación del arduino un valor de 20%. La documentación externa y el anexo tienen un valor de 20%.
8. De las notas mencionadas en el punto anterior se calculará la Nota Final del Proyecto.
9. Se evaluará que la documentación sea coherente, acorde a la dificultad/tamaño del proyecto y el trabajo realizado. Se recomienda que realicen la documentación conforme se implementa el código.
10. La nota de la documentación externa es proporcional a la completitud del proyecto.
11. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
12. Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la primera cita de revisión oficial.
13. Aun cuando el código y la documentación externa tienen sus notas por separado, se aplican las siguientes restricciones
 - a. Si no se entrega documentación externa, automáticamente se obtiene una nota de cero en la nota final del proyecto.
 - b. Si no se utiliza un manejador de código se obtiene una nota de cero en la nota final del proyecto.
 - c. Si la documentación externa no se entrega en la fecha indicada se obtiene una nota de cero en la nota final del proyecto.
 - d. Si el código no compila se obtendrá una nota de cero en la nota final del proyecto, por lo cual se recomienda realizar la defensa con un código funcional.
 - e. El código debe ser desarrollado en Java (Windows), en caso contrario se obtendrá una nota de cero en la nota final del proyecto.
 - f. Si alguna persona integrante del proyecto no se presenta a la revisión se asignará una nota de cero en la nota final del proyecto.
14. La revisión de la documentación podrá ser realizada antes, durante o después de la revisión del proyecto.
15. Cada estudiante tendrá como máximo 30 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
16. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
17. Cada estudiante 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 al profesor para coordinar el préstamo de estos.
18. Durante la revisión únicamente podrán participar el estudiante, asistentes, otros profesores y el coordinador del área.

ANEXO DEL PROYECTO (2.5% Adicional)

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.
- Validar 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

- Diseño (Inicial).

Descripción del Entregable

Cada grupo debe elaborar un documento 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. Describir el diseño implementado (diagrama de clases) y una alternativa de diseño diferente a la solicitada en la especificación del proyecto (diagrama de arquitectura). Para esto se recomienda considerar que este sistema fuera desarrollado para resolver un problema de la vida real.
 - c. Describa, mediante un checklist, las funcionalidades que fueron implementadas y las que quedaron pendientes.

Aspectos operativos y evaluación

1. El documento debe ser enviado en formato PDF, junto con los demás entregables del proyecto.