
Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

Algoritmos y Estructuras de Datos I

II Semestre 2024 Proyecto III: AirWar

Josue Venegas Arias 2023075124

Alejandro Arias Alfaro 2022438568

Tabla de Contenidos

1. Introducción
 2. Diseño
 - a. Listado de requerimientos descritos como historias de usuario
 - b. Alternativas de solución para problemas complejos
 - c. Diagramas UML y de arquitectura
 - d. Checklist de historias de usuario implementadas
 3. Resultados
 4. Conclusión
-

Introducción

El proyecto **AirWar** es un videojuego desarrollado en C#, que simula un escenario de guerra aérea con múltiples elementos dinámicos y sistemas autónomos. Este proyecto tiene como propósito principal aplicar conceptos de programación orientada a objetos y estructuras de datos avanzadas, como grafos, listas de adyacencia y algoritmos de búsqueda y ordenamiento, para crear un entorno interactivo, funcional y educativo.

El desarrollo de **AirWar** permite a los estudiantes profundizar en la implementación de técnicas de programación mientras resuelven problemas de diseño y lógica. Además, fomenta el uso de herramientas modernas como Unity, para ofrecer una experiencia de usuario profesional e intuitiva.

Se busca también simular módulos de "inteligencia artificial" dentro del juego, destacando la versatilidad del software para modelar comportamientos complejos de manera eficiente.

Diseño

a. Listado de Requerimientos Descritos como Historias de Usuario

Requerimiento 000: Selección del Entorno de Desarrollo

- **Descripción:** El juego debe desarrollarse en C#, utilizando frameworks modernos como Unity,
- **Propósito:** Garantizar un entorno de desarrollo robusto y versátil que permita crear gráficos atractivos y manejar eventos complejos.
- **Implementación:** Se eligió Unity debido a su compatibilidad con C#, su motor gráfico avanzado y su soporte para simulaciones en tiempo real.

Requerimiento 001: Objetivo del Juego

- **Descripción:** El objetivo es destruir la mayor cantidad de aviones en un periodo de tiempo limitado.
- **Propósito:** Proveer un desafío central que incentive al jugador a interactuar con el sistema de manera estratégica.
- **Implementación:** Se estableció un temporizador global y un contador de aviones derribados para rastrear el progreso del jugador.

Requerimiento 002: Generación Aleatoria de Elementos y Rutas

- **Descripción:** Generar aeropuertos, portaaviones y rutas aleatoriamente, utilizando un grafo representado mediante listas de adyacencia.
- **Propósito:** Crear un mapa dinámico y variado en cada partida, mejorando la jugabilidad del juego.
- **Implementación:** Se utilizó un algoritmo personalizado para generar grafos aleatorios.

Requerimiento 003: Movimiento de la Batería Antiaérea

- **Descripción:** El jugador controla una batería antiaérea que se mueve horizontalmente con velocidad constante y dispara balas con trayectoria recta.
- **Propósito:** Ofrecer una mecánica interactiva que requiera habilidad y precisión.
- **Implementación:** Se usaron animaciones en Unity para el movimiento de la batería y un sistema de cálculo de velocidad variable para las balas, dependiendo del tiempo que el jugador mantiene presionado el disparador.

Requerimiento 004: Rutas con Pesos Personalizados

- **Descripción:** Las rutas deben tener un peso calculado según la distancia, el tipo de destino y el tipo de ruta.

- **Propósito:** Agregar un nivel de realismo y complejidad al sistema de navegación aérea.
- **Implementación:** Se modificó el algoritmo de Dijkstra para incluir múltiples factores en el cálculo del peso de las rutas.

Requerimiento 005: Comportamiento Autónomo de los Aviones

- **Descripción:** Los aviones deciden destinos aleatorios y calculan la mejor ruta, aterrizan, recargan combustible y vuelven a despegar.
- **Propósito:** Simular un entorno realista donde los aviones operen de manera independiente.
- **Implementación:** Cada avión tiene un módulo que simula decisiones autónomas, incluyendo un temporizador para simular el tiempo de espera en el aeropuerto.

Requerimiento 006: Gestión de Combustible

- **Descripción:** Los aeropuertos deben racionar combustible según su capacidad, y los aviones pueden caer si no tienen suficiente.
- **Propósito:** Introducir un elemento estratégico que simule recursos limitados en un sistema dinámico.
- **Implementación:** Se utilizó una cola de prioridad para asignar combustible a los aviones en orden de llegada.

Requerimiento 007: Creación de Aviones

- **Descripción:** Los aeropuertos generan nuevos aviones, limitados por la capacidad de sus hangares.
- **Propósito:** Asegurar un flujo constante de elementos dinámicos en el juego.
- **Implementación:** Los aviones tienen IDs únicos generados con GUIDs y se crean cuando el hangar tiene espacio disponible.

Requerimiento 008: Simulación de Módulos de AI

- **Descripción:** Cada avión contiene cuatro módulos que trabajan juntos: Pilot, Copilot, Maintenance y Space Awareness.
- **Propósito:** Modelar la autonomía de los drones mediante una simulación detallada.
- **Implementación:** Los módulos tienen atributos como ID, rol y horas de vuelo. La interacción entre módulos se simula mediante eventos programados.

Requerimiento 009: Ordenamiento de Datos

- ## Requerimiento 010: Visualización de Datos

- ### **b. Alternativas de Solución para Problemas Complejos**

Problema: Rutas con Pesos Personalizados

- # Diagramas UML y Arquitectura



Resultados

- **Implementación Exitosa:** Se lograron la mayoría los requerimientos básicos y se optimizó el rendimiento en tiempo real.
 - **Interfaz Gráfica:** La visualización de datos se ejecuta correctamente y se actualiza dinámicamente.
-

Conclusión

El proyecto **AirWar** cumplió con los objetivos de aplicar programación orientada a objetos y estructuras de datos avanzadas. Además, introdujo simulaciones complejas de comportamientos autónomos, demostrando la viabilidad de modelar sistemas complejos de manera eficiente.
