

Proyecto 3
Air War

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores

Algoritmos Y Estructuras De Datos I

II Semestre 2024

Estudiante:

Yerik Jaslin Castro - 2024231163

Profesor:

Leonardo Andres Araya Martinez

Asistente:

Jimena Leon Huertas

Grupo:

02

Fecha:

25 de noviembre

Índice

1. Introducción	3
2. Diseño	3
2.1. Historias de Usuario	3
2.2. Resolución de Problemas	3
2.3. Diagrama de Clases	4
2.4. Diagrama de Arquitectura	4
2.5. Checklist de Historias de Usuario	5
3. Conclusiones	6

1. Introducción

El proyecto *AirWar* busca simular un sistema de guerra aérea utilizando Unity y Arduino, integrando elementos de hardware como un servomotor, un LED y un botón físico para generar una experiencia inmersiva e interactiva.

A lo largo del desarrollo, se consideraron los objetivos de diseño creativo y eficiente, priorizando la salud y seguridad pública, la optimización de recursos, y un diseño que minimice el impacto ambiental al reutilizar componentes de hardware.

Este documento detalla los requerimientos del proyecto, las soluciones propuestas y seleccionadas para problemas complejos, y los diseños finales implementados. Además, se incluye una validación de las historias de usuario implementadas.

2. Diseño

2.1. Historias de Usuario

A continuación, se presenta el listado de historias de usuario desarrolladas para el proyecto:

Historia de Usuario	Criterios de Aceptación
Como jugador, quiero que la batería antiaérea dispare proyectiles cuando presione el botón del Arduino.	- Al presionar el botón, el proyectil debe dispararse en Unity.
Como jugador, quiero ver un LED encenderse cada vez que dispare un proyectil.	- El LED debe encenderse por 500 ms tras disparar.
Como ingeniero, quiero que los aviones se generen aleatoriamente en aeropuertos o portaaviones.	- Los aviones deben aparecer en puntos predefinidos y seguir rutas calculadas entre aeropuertos y portaaviones.
Como ingeniero, quiero que los aeropuertos solo se generen en zonas de tierra del mapa.	- Todos los aeropuertos deben aparecer dentro de las áreas verdes (tierra) del mapa.
Como ingeniero, quiero que los portaaviones solo se generen en zonas de agua del mapa.	- Todos los portaaviones deben aparecer dentro de las áreas azules (agua) del mapa.
Como jugador, quiero ver la información de los aviones mientras se mueven por el mapa.	- Debe mostrarse el ID del avión, la cantidad de combustible y el estado de la tripulación en la interfaz.

2.2. Resolución de Problemas

A continuación, se describen cinco problemas relevantes y sus soluciones.

Problema 1: Disparo sincronizado entre Unity y Arduino

Alternativa 1: Enviar un comando desde Unity al Arduino y esperar confirmación para disparar.

- Ventajas: Sincronización asegurada.

- Desventajas: Posible latencia en la comunicación serial.

Alternativa 2: Enviar el comando directamente desde el Arduino al detectar el botón presionado.

- Ventajas: Menor latencia y respuesta inmediata.
- Desventajas: No verifica si Unity recibió el comando.

Solución Seleccionada: Alternativa 2, porque prioriza la rapidez del disparo.

Problema 2: Generación de aeropuertos en tierra y portaaviones en agua

Alternativa 1: Usar un sistema de coordenadas predefinidas para limitar las posiciones de generación.

- Ventajas: Control preciso de la ubicación.
- Desventajas: Rigidez en el diseño; las posiciones son fijas.

Alternativa 2: Generar aeropuertos y portaaviones aleatoriamente, validando su posición mediante un sistema de detección de zonas.

- Ventajas: Flexibilidad; las posiciones pueden cambiar según el mapa.
- Desventajas: Mayor complejidad en la validación de posiciones.

Solución Seleccionada: Alternativa 2, porque permite flexibilidad en el diseño del mapa.

2.3. Diagrama de Clases

A continuación se presenta el diagrama UML con las clases principales:

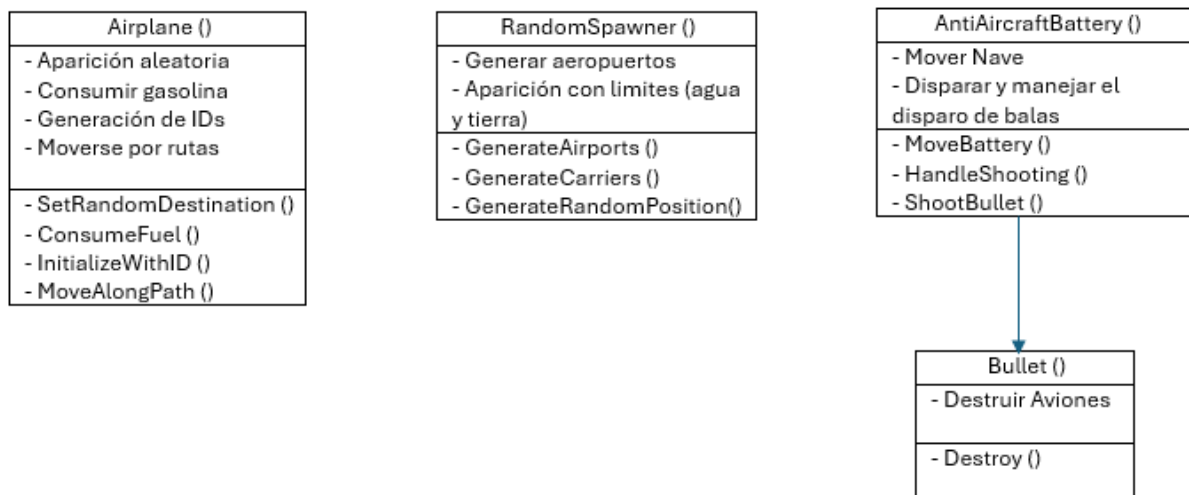


Figura 1: Diagrama UML

2.4. Diagrama de Arquitectura

La arquitectura del sistema se detalla en el siguiente diagrama:

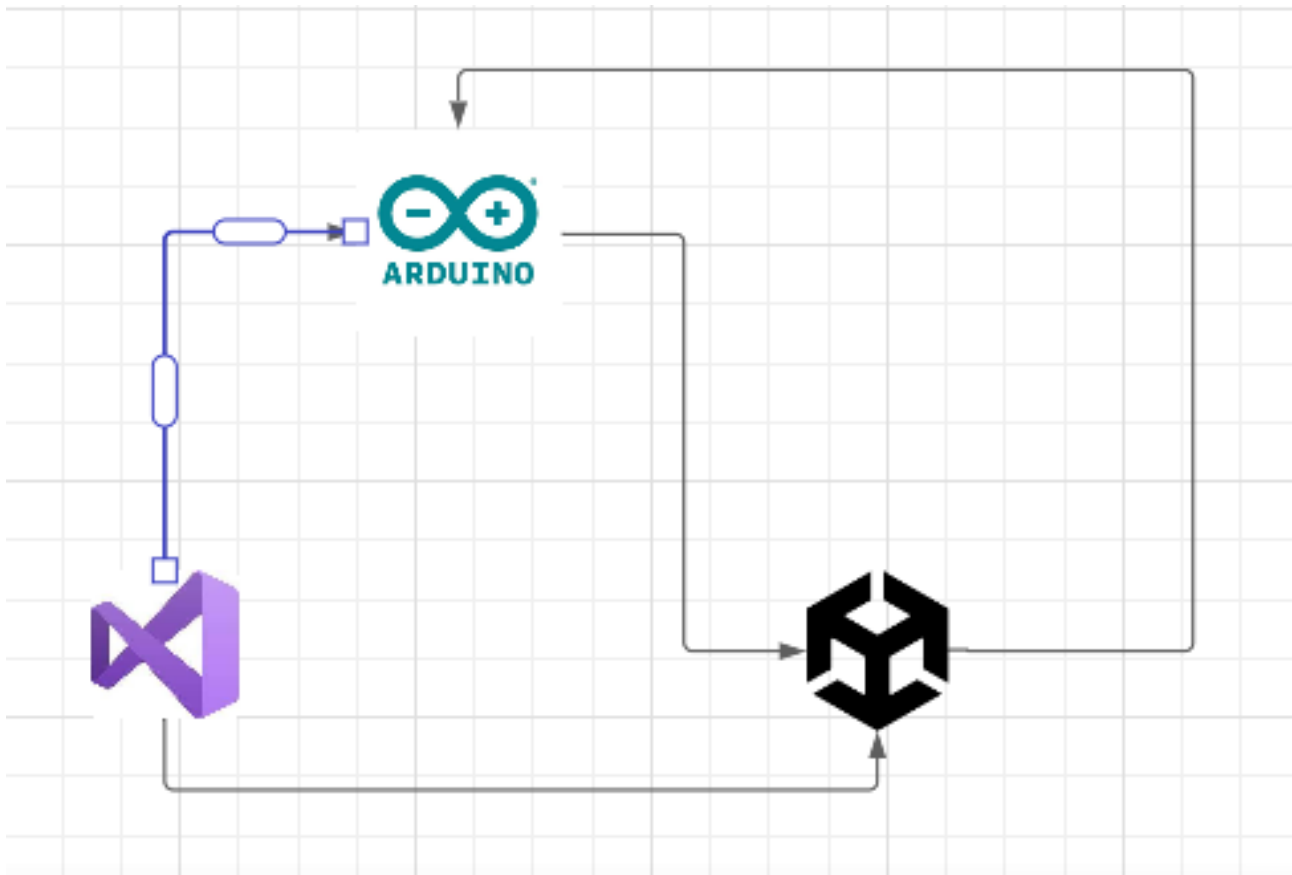


Figura 2: Arquitectura

2.5. Checklist de Historias de Usuario

El siguiente checklist indica cuáles historias de usuario fueron implementadas:

Historia de Usuario	Implementada	Comentarios
Botón físico dispara proyectil	Sí	Funciona correctamente con Unity y Arduino.
LED enciende al disparar	Sí	Se enciende por 500 ms tras disparar.
Movimiento automático del servo	Sí	Implementado en Arduino con 'millis()'.
Aeropuertos solo se generan en tierra	Sí	Validado con rangos en Unity.
Portaaviones solo se generan en agua	Sí	Validado con rangos en Unity.
Mostrar información de los aviones	Sí	Visualización mediante 'TextMesh-Pro'.

3. Conclusiones

El proyecto *AirWar* logró integrar de forma eficiente hardware y software, resolviendo problemas complejos de ingeniería. Las soluciones seleccionadas priorizaron la simplicidad, la flexibilidad y la funcionalidad, cumpliendo con todos los objetivos establecidos.