

Rogue Drones

Victor Montes Garcia

Máster Universitario en Diseño y Programación de Videojuegos
Trabajo Final de Máster

Helio Tejedor Navarro
Joan Arnedo Moreno

07/06/2020



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Rogue Drones</i>
Nombre del autor:	<i>Victor Montes Garcia</i>
Nombre del consultor/a:	<i>Helio Tejedor Navarro</i>
Nombre del PRA:	<i>Joan Arnedo Moreno</i>
Fecha de entrega (mm/aaaa):	06/2020
Titulación:	<i>Máster Universitario en Diseño y Programación de Videojuegos</i>
Área del Trabajo Final:	<i>Trabajo Final del Máster</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Videojuegos, Estrategia, Unity</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>El objetivo de este trabajo ha sido poner en práctica todo lo aprendido a lo largo de este máster. Para ello se ha decidido realizar un juego de estrategia táctica en tiempo real con el motor Unity 3D. Este juego consiste en controlar un dron que tiene la posibilidad de capturar a las unidades y estructuras del enemigo y usarlas para poder llegar al final del nivel. En este juego se han volcado los conocimientos adquiridos y se han implementado desarrollando diversos aspectos del juego, como su jugabilidad, diseño de niveles, efectos visuales y sonoros, físicas, inteligencia artificial e interacción del jugador. Para realizar este trabajo se han ido marcando una serie de objetivos marcados a lo largo del tiempo establecido para su desarrollo, al fin de llegar a su cumplimiento en las diversas entregas establecidas por el programa de estudios. El alumno no se propuso un objetivo ambicioso por las limitaciones de tiempo, y se puso unas metas acordes a este. Estas metas se han alcanzado en el tiempo estipulado. A modo de conclusión ha quedado claro que el alumno ha sacado provecho del conocimiento adquirido, entregando un juego jugable con las características esperadas, aunque, también es cierto que las limitaciones de tiempo y la falta de conocimiento en algunas partes específicas del desarrollo han hecho que el juego no destaque fuera de un producto destinado a ser evaluado académicamente, aunque sí puede servir de base para un producto final en un futuro.</p>	

Abstract (in English, 250 words or less):

The objective of this project has been to put into practice everything learned throughout this master's degree. For this purpose, it has been decided to make a real time tactical strategy game with the Unity 3D engine. This game consists of controlling a drone that is able of capturing the units and structures of the enemy and use them to reach the end of the level. In this game, the knowledge acquired has been applied by developing various aspects of the game, such as its gameplay, level design, visual and sound effects, physics, artificial intelligence, and player interaction. To do this work, a series of objectives have been set throughout the time established for its execution, to reach its accomplishment in the different deliveries established by the study program. The student did not set an ambitious objective due to time constraints, and the goals were set accordingly. These goals have been achieved in the stipulated time. As a conclusion, it is clear that the student has taken advantage of the knowledge acquired, delivering a playable game with the expected characteristics, although it is also true that time limitations and lack of knowledge in some specific parts of the development have made the game not stand out from a product destined to be evaluated academically, although it can serve as a basis for a final product in the future.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	3
1.4 Planificación del Trabajo.....	3
1.4.1 PEC 2 - Versión parcial (Inicio: 02/03/20 - Entrega: 05/04/20).....	4
1.4.2 PEC 3 - Versión jugable (Inicio: 06/04/20 - Entrega: 24/05/20).....	4
1.4.3 PEC Final - Versión final (Inicio: 25/05/20 - Entrega: 07/06/20).....	5
1.5 Breve resumen de productos obtenidos.....	5
1.6 Breve descripción de los otros capítulos de la memoria.....	6
1.6.1 Capítulo 2: Estado del arte.....	6
1.6.2 Capítulo 3: Definición del juego.....	6
1.6.3 Capítulo 4: Diseño técnico.....	6
1.6.4 Capítulo 5: Diseño de niveles.....	6
1.6.5 Capítulo 6: Manual de usuario.....	6
2. Estado del arte.....	7
2.1 Género.....	7
2.2 Análisis del género.....	7
2.3 Plataformas de desarrollo empleadas.....	10
3. Definición del juego.....	11
3.1 Historia.....	11
3.2 Interacción entre los actores del juego.....	11
3.3 Definición de los personajes / elementos.....	14
3.4 Objetivos planteados al jugador.....	19
4. Diseño técnico.....	20
4.1 Entorno de desarrollo.....	20
4.2 Requerimientos técnicos.....	20
4.3 Herramientas empleadas.....	20
4.3 Assets y recursos al juego.....	21
4.4 Esquema de arquitectura del juego / componentes.....	24
4.4.1 Drones.....	24
4.4.1.1 BasicDrone.....	24
4.4.1.2 DroneInterface.....	25
Clases específicas de un dron.....	25
4.4.1.3 AirDrone.....	25
4.4.1.4 BomberDrone.....	25
4.4.1.5 HealerDrone.....	25
4.4.1.6 MainDrone.....	26
4.4.1.7 PEMDrone.....	26
4.4.1.8 ScoutDrone.....	26
4.4.1.9 TankDrone.....	26
4.4.2 Armas.....	26
4.4.2.1 Bomb.....	26
4.4.2.2 Missile.....	27
4.4.2.3 Missile_Collider.....	27

4.4.2.4 Missile_Launcher.....	27
4.4.3 Estructuras.....	27
4.4.3.1 BasicStructure	27
4.4.3.2 StructuresInterfaces.....	27
Clases específicas de una estructura	28
4.4.3.4 AA_Gun	28
4.4.3.5 AA_Rotation.....	28
4.4.3.6 Armory	28
4.4.3.7 Barracs	28
4.4.3.8 CommandPost	28
4.4.3.9 GatlingGunCustom	29
4.4.3.10 Generator.....	29
4.4.3.11 PEMTower	29
4.4.3.12 Radar	29
4.4.4 Jugabilidad.....	30
4.4.4.1 AuxiliarOperation	30
4.4.4.2 ColliderCommunicator	30
4.4.4.3 CommonInterface	30
4.4.4.4 GameConstant.....	30
4.4.4.5 EndLevel.....	30
4.4.4.6 FogOfWar	31
4.4.4.7 GameplayManager	31
4.4.4.8 Tutorial.....	31
4.4.4.9 ApplicationManager	31
4.4.5 Jugador.....	31
4.4.5.1 PlayerMovement.....	31
4.4.6 Interfaz gráfica	32
4.4.6.1 HUDManager.....	32
4.4.6.2 HUDPlayerDrone	32
4.4.6.3 TutorialScript.....	32
4.4.7 Shaders.....	32
4.4.7.1 FogOfWar	32
4.4.8 Cámara	33
4.4.8.1 RTSCam	33
4.4.9 Escenarios de test	33
4.5 IA de los enemigos.....	34
4.5.1 Estructuras.....	34
4.5.2 Drones	34
4.5.2.1 Patrulla.....	36
4.5.2.2 Ataque	36
4.5.2.3 Alerta	37
4.5.2.4 Capturado	37
5. Diseño de niveles	38
5.1 Tutorial	38
5.2 Nivel de juego.....	40
6. Manual de usuario.....	43
6.1 Controles	43
6.2 HUD.....	43
6.3 Menús.....	44
6.3.1 Menú de inicio	44

6.3.2 Menú de pausa	44
6.3.3 Menú de fin del juego.....	45
6.4 Desarrollo de la partida	46
6.4.1 Mover drones	46
6.4.2 Captura de drones y estructuras.....	46
6.4.3 Ataque.....	46
6.4.4 Recuperación de salud y munición	47
6.4.5 Ganar la partida	47
6.4.6 Fin de la partida	48
6.5 Requisitos mínimos	49
7. Conclusiones.....	50
7.1 Lecciones aprendidas.....	50
7.2 Reflexión sobre los objetivos planteados	50
7.2.1 Inteligencia Artificial	51
7.2.2 Drones	51
7.2.3 Infraestructuras	51
7.2.4 Mecánicas jugables	51
7.2.5 Diseño de niveles.....	51
7.2.6 Control de los personajes e interacción con el juego	52
7.2.7 Flujo del juego.....	52
7.2.8 Interfaz de usuario	52
7.2.9 Documentación	52
7.2.10 Tareas de refinamiento / corrección.....	52
7.3 Sobre la planificación y metodología seguida	52
7.4 Líneas de trabajo futuro.....	53
8. Glosario	54
9. Bibliografía	55
10. Anexos	56
10.1 Problemas durante el desarrollo.....	56

Lista de figuras

Ilustración 1	Imágenes de los juegos tomados como referencia	8
Ilustración 2	Muestra de la zona de cobertura en el escenario de juego.	12
Ilustración 3	Dron principal capturando otro dron.	12
Ilustración 4	Torre de repetición de señales	13
Ilustración 5	Puesto de mando	13
Ilustración 6	Torre de inhibición y su generador.	14
Ilustración 7	Collider de una torre artillada.	34
Ilustración 8	Diagrama de flujo de los estados de la IA	35
Ilustración 9	Collider de un dron explorador	36
Ilustración 10	Vista aérea del nivel de tutorial	38
Ilustración 11	Vista aérea del nivel del juego	40
Ilustración 12	Captura del HUD durante la partida.	43
Ilustración 13	Captura del menú de inicio	44
Ilustración 14	Captura del menú de pausa	45
Ilustración 15	Captura del menú de fin del juego	45
Ilustración 16	Captura de un puesto de mando	46
Ilustración 17	Dron de ataque atacando a un arma defensiva	47
Ilustración 18	Centro de reparación reparando el dron principal	47
Ilustración 19	imágenes de las zonas marcadas como fin de nivel	48

1. Introducción

1.1 Contexto y justificación del Trabajo

El objetivo de este trabajo de fin de máster es entregar un juego funcional que demuestre los conocimientos que el alumno ha adquirido a lo largo de este máster.

El alumno ha decidido desarrollar un juego de estrategia táctica, gracias al cual se podrá poner a prueba los diversos conocimientos adquiridos en el ámbito del desarrollo de videojuegos, los cuales van desde conocimientos en programación, jugabilidad, inteligencia artificial, efectos visuales y sonoros y diseño de escenarios y niveles jugables.

El objetivo de este juego consiste en controlar un dron que tiene la posibilidad de capturar a las unidades y estructuras del enemigo y usarlas para poder llegar al final del nivel. El desarrollo del juego y los elementos que lo componen se expondrán a lo largo de este documento.

1.2 Objetivos del Trabajo

Como se ha explicado en el anterior apartado, el juego que queremos desarrollar debe ser completamente funcional, por lo que es necesario desarrollar todos los elementos necesarios para garantizar una experiencia jugable.

Estos elementos no van solo de los mencionados anteriormente como pueden ser la IA, la programación o el escenario, también se tienen que implementar elementos que se dan por supuestos en un juego como los personajes, enemigos, interfaz de control o el sonido.

A destacar elementos que se mencionan a lo largo de este documento como son los drones, los cuales son la base jugable pues serán los elementos que controlaremos, y las estructuras, que darán soporte o dificultan la tarea a estos últimos.

Para poder desarrollar a tiempo el proyecto, ha sido necesario dividirlo en diferentes subtareas. Algunas abarcan aspectos comunes a todo el juego mientras otras solo afectan a actores concretos de este, e incluso algunas de ellas podrían llegar a considerarse como extras, dependiendo estas últimas del tiempo de desarrollo.

A continuación, se enumeran las diferentes subtareas y los objetivos asociados a ellas:

1. Inteligencia Artificial:
 - a. Máquina de estados de los enemigos

- b. Pathfinding (navegación)
 - c. Mecánicas de ataque
 - d. Ajuste de comportamiento para cada Dron
- 2. Drones:
 - a. Búsqueda y diseño de Assets
 - b. Animación
 - c. Sonido
 - d. Gestión de atributos (salud, munición, coste pirateo ...)
 - e. Desarrollo de un tipo de cada dron:
 - 1. Dron principal
 - 2. Drones Aéreos
 - 3. Drones terrestres
 - 4. Drones de operaciones especiales
 - 5. Drones de apoyo auxiliar
- 3. Infraestructuras:
 - a. Búsqueda y diseño de Assets
 - b. Animación
 - c. Sonido
 - d. Desarrollo de un tipo de cada infraestructura:
 - 1. Repetidores de señal
 - 2. Inhibidores de señal
 - 3. Estructuras defensivas
 - 4. Puestos de mando móvil
 - 5. Centros de reparación
 - 6. Polvorines
 - 7. Infraestructuras de soporte
- 4. Mecánicas jugables:
 - a. Pirateo de drones
 - b. Pirateo de infraestructuras
 - c. Recuperación y pérdida de salud y munición
 - d. Zona jugable
- 5. Diseño de niveles:
 - a. Escenario de demo
 - b. Diseño del escenario
 - c. Rutas de los enemigos
 - d. Ubicación de los enemigos y las infraestructuras
- 6. Control de los personajes e interacción con el juego
- 7. Flujo del juego:
 - a. Definición de pantallas principales:
 - 1. Título
 - 2. Juego
 - 3. Pausa
 - 4. Opciones
 - 5. Resultados
 - 6. Game over
 - b. Flujo entre las diferentes pantallas
 - c. Condiciones de victoria y derrota
- 8. Interfaz de usuario:
 - a. HUD de información
 - b. Menús de configuración de la partida

9. Documentación:
 - a. Informe PECs
 - b. Redacción memoria
 - c. Creación de vídeo explicativo
10. Tareas de refinamiento / corrección:
 - a. Refinamiento IA
 - b. Refinamiento Jugabilidad
 - c. Corrección bugs
11. Tareas opcionales:
 - a. Implementación de más drones
 - b. Mecánicas enfocadas a la narrativa
 - c. Confrontación entre las 2 facciones del juego

El desarrollo de estas tareas ayudara a entender la planificación expuesta en el punto **1.4 Planificación del Trabajo**.

1.3 Enfoque y método seguido

El TFM se dividió entre diversas entregas, en cada una de ellas se ponía como objetivo el desarrollo y la finalización de las diferentes subtareas y objetivos que los componen.

El objetivo era tener una versión que facilitase el tiempo de dedicación a la siguiente entrega, pues el tiempo era limitado y la posibilidad de que surgieran inconvenientes en el desarrollo era algo que se quería tener bajo control.

Este método de trabajo, junto con la planificación efectuada, convierte el desarrollo en una metodología ágil, aunque no entraría en el marco de un tipo específico, pues el alumno fue tanto el desarrollador como el propietario del producto.

Cada una de las entregas tuvo un periodo de tiempo distinto para ser completada, pero el alumno sólo disponía de un tiempo limitado por semana para desarrollar los objetivos propuestos. Para ejemplificar esto, e indicar el tiempo que se iba a dedicar a cada tarea, se hizo una serie de diagramas de Gantt para cada una de las entregas, los cuales son explicados en el siguiente apartado.

1.4 Planificación del Trabajo

Como se indicó en el anterior apartado, para cada una de las diferentes entregas de este trabajo de fin de máster se planeó una planificación que queda ilustrada con los siguientes diagramas de Gantt.

Es importante indicar que, si bien en la práctica la planificación no se ha seguido al 100%, el itinerario seguido a sido muy similar, variando algunas fechas o tiempo dedicado.

1.4.1 PEC 2 - Versión parcial (Inicio: 02/03/20 - Entrega: 05/04/20)

El objetivo marcado para esta entrega era tener una implementación de un dron para cada tipo, así como cada tipo de infraestructura.

Para probarlo se hizo un pequeño escenario de demo para comprobar de forma individual cada uno de los elementos.

El diseño de estos elementos implicó la implementación de los elementos comunes para todos ellos como las mecánicas de salud y munición o el control de estos elementos cuando lo controle el jugador.



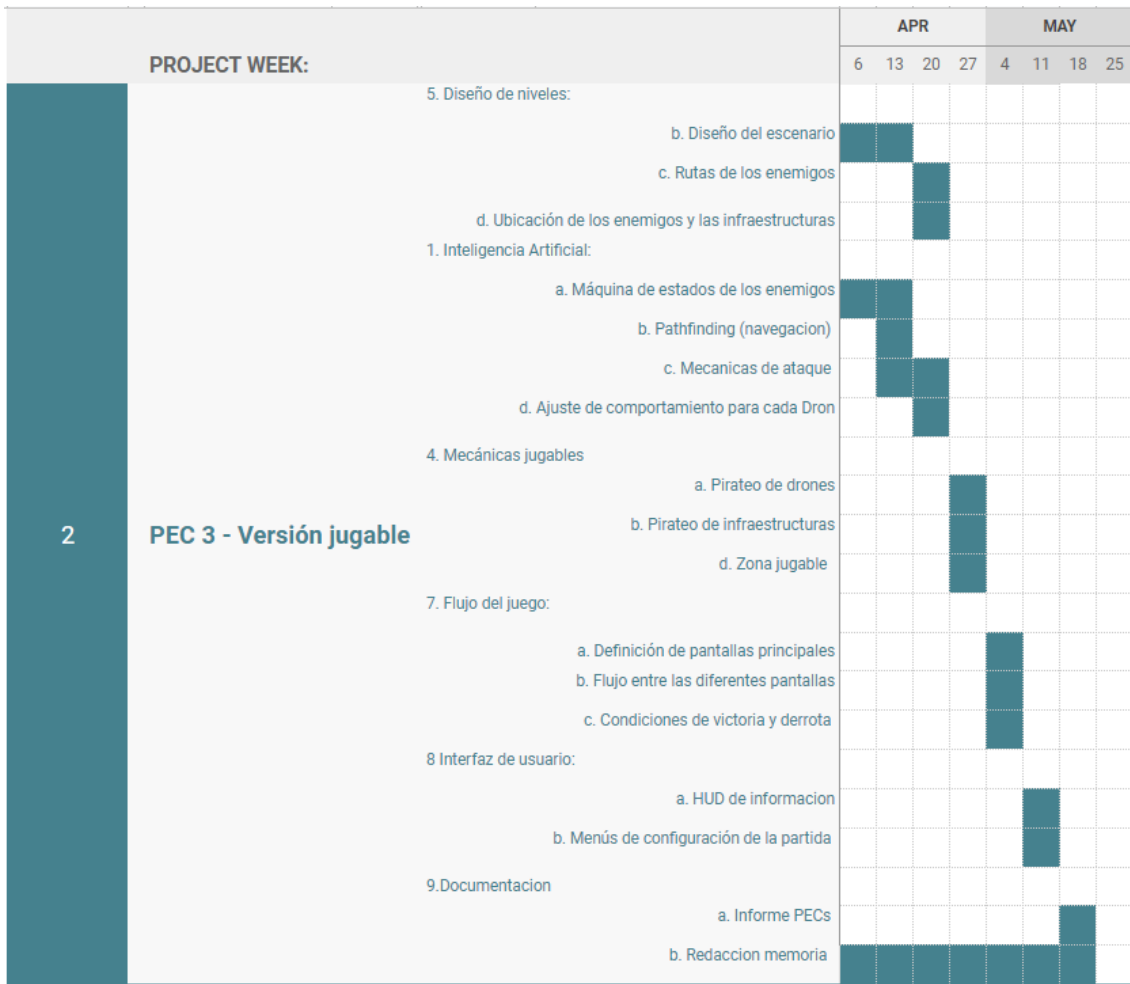
1.4.2 PEC 3 - Versión jugable (Inicio: 06/04/20 - Entrega: 24/05/20)

En esta entrega se desarrolló el escenario final del proyecto. El objetivo era poder unir todos los elementos que se llegaron a desarrollar en la entrega anterior.

Al unir estos elementos fue necesario desarrollar las mecánicas jugables que no se podían desarrollar con los elementos por separado, como el pirateo de drones o la interacción entre jugador y múltiples enemigos (ver apartado de definición del juego).

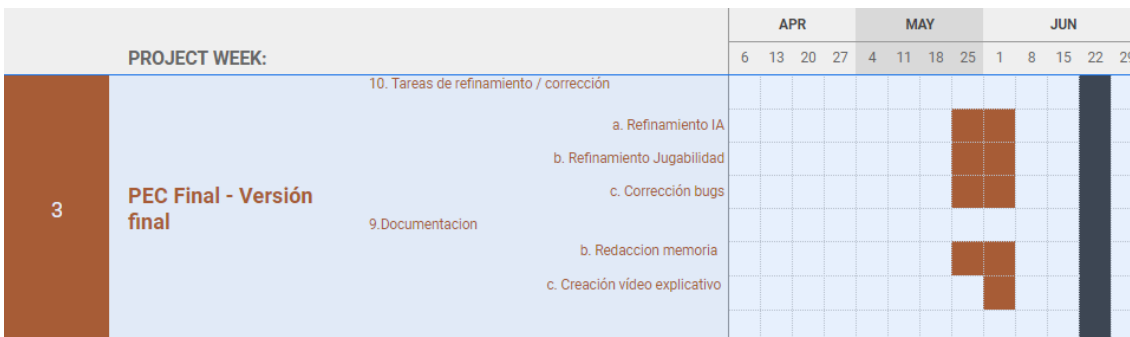
También se desarrolló la IA, los objetivos de victoria y derrota del jugador y el resto de las pantallas del juego y el flujo entre estas.

En esta entrega se empezó redactar la memoria del proyecto, con el objetivo de no dejarla en su totalidad en la última entrega



1.4.3 PEC Final - Versión final (Inicio: 25/05/20 - Entrega: 07/06/20)

El objetivo de esta última entrega era acabar de perfilar el proyecto entregable, así como finalizar la documentación final de la asignatura.



1.5 Breve resumen de productos obtenidos

El objetivo de entregar un juego jugable se ha cumplido durante el desarrollo de este trabajo de fin de máster, así como la diferente documentación solicitada.

En total el alumno ha entregado lo siguiente:

- Videojuego Rogue Drones completamente jugable (ver repositorio: https://github.com/vimoga/Trabajo_Fin_Master con la tag PEC_FINAL)
- Video presentación describiendo el desarrollo y jugabilidad del juego (<https://youtu.be/FrbDF5GfRRQ>)
- Memoria del proyecto de fin de máster
- Unity Showcase del juego Rogue Drones (<https://connect.unity.com/p/rogue-drones>)
- Documentación y videos para cada una de las entregas del TFM
- Gameplay del escenario de tutorial del juego (<https://youtu.be/wqb0IRtbFvk>)
- Gameplay del escenario de juego (<https://youtu.be/w79SPWchrRE>)

1.6 Breve descripción de los otros capítulos de la memoria

1.6.1 Capítulo 2: Estado del arte

Incluye una definición del género del juego, así como una comparativa con otros juegos del mismo estilo. También se indica con qué tipo de tecnología se diseñaron estos juegos.

1.6.2 Capítulo 3: Definición del juego

Explicación del objetivo del juego, los diversos elementos que lo componen y la interacción con el jugador.

1.6.3 Capítulo 4: Diseño técnico

Incluye una explicación del entorno de desarrollo escogido, un inventario de las herramientas y assets usados en el juego y una explicación de las diferentes clases que componen la lógica del juego.

También se expone el funcionamiento de la IA.

1.6.4 Capítulo 5: Diseño de niveles

Se expone los niveles implementados y las posibilidades/opciones del jugador.

1.6.5 Capítulo 6: Manual de usuario

Se explica las instrucciones del juego y requerimientos mínimos

2. Estado del arte

2.1 Género

El género que vamos a desarrollar en este TFM es el de táctica en tiempo real (Real Time Tactics). A diferencia de un género de estrategia en tiempo real, donde además del control de diferentes unidades intervienen otros aspectos como la construcción o la gestión y obtención de recursos, este género hace enfoque en las unidades en sí, dándoles habilidades únicas para sortear los diferentes obstáculos propuestos, siendo el jugador el que tiene que pensar que táctica y que unidad/es es la adecuada para cada situación [4].

2.2 Análisis del género

En el género de la táctica en tiempo real hay varios referentes, siendo el principal la saga Commandos (Pyro Studios), siendo considerada por muchos como la fundadora del género. Esta saga tuvo su competencia con la franquicia Desperados (Spellbound Entertainment), pero hay que remarcar que ambas son de principios de los 2000. Desde entonces ha habido pocas incursiones en el género hasta la llegada de Shadow tactics: blades of the shogun (Mimimi Productions) en 2016. Actualmente se está relanzando este tipo de juegos, con nuevos lanzamientos de las franquicias antes mencionadas y remasterizaciones.





Ilustración 1 Imágenes de los juegos tomados como referencia

A la hora de comparar nuestro juego se hará con el anteriormente mencionado Shadow Tactics Blades of the shogun (2016) y los juegos más característicos de las sagas anteriormente mencionadas, Commandos 2: men of courage (2001) y Desperados: wanted dead or alive (2001). Poner en relieve que se está haciendo la comparación con los elementos que se han desarrollado para la entrega de este TFM, no para un posible juego final.

Para ello se definirán las características que definen nuestro juego:

- C1: Posibilidad de controlar varios personajes con habilidades únicas.
- C2: Mapa abierto (el jugador no tiene más límite que las limitaciones geográficas para moverse por el escenario).
- C3: Escenarios con objetivos principales y secundarios.
- C4: Es posible cumplir los objetivos definidos de varias maneras.
- C5: Mecánicas de combate y sigilo.
- C6: Existen caminos o posibilidades de cumplir los objetivos que están limitados a interacciones con el entorno.
- C7: Posibilidad de escoger entre múltiples escenarios (niveles) entre misión y misión.
- C8: La inteligencia artificial (y diseño de niveles) busca siempre romper la estrategia preconcebida del jugador.
- C9: El entorno narrativo y jugable busca una representación cruda de un conflicto bélico.
- C10: Posibilidad de gestionar el equipo (ítems) de los personajes jugables.
- C11: Es necesario que el jugador cumpla requisitos para llegar a realizar ciertas misiones.

Se puede contemplar las diferencias entre los distintos juegos en la siguiente tabla:

Juego / Característica	Rogue Drones	Desperados	Commandos 2	Shadow tactics
C1	X	X	X	X
C2	X	X	X	X
C3			X	X
C4	X	X	X	X
C5	X	X	X	X
C6	X		X	X
C7				X
C8	X	X	X	X
C9			X	
C10		X	X	X
C11			X	

Como se puede ver, se comparten ciertas características con los juegos antes mencionados, sin embargo, la falta de tiempo y medios que implica el desarrollo del TFM impide crear en su estado actual un juego que destaque sobre la competencia. Aunque es necesario indicar que el alumno cree que existe suficiente potencial no solo para añadir las mismas características que sus competidores, si no mejorándolas, incluyendo aspectos como una mejor IA, apartado artístico o la inclusión de elementos narrativos.

2.3 Plataformas de desarrollo empleadas

En los ejemplos proporcionados, en el caso de los juegos Commandos 2 se usó un motor gráfico propio, aunque, al poco de escribir esta memoria se hizo un remaster de este en el que se empleó el motor Unity [6]. Este motor también se usó para el juego Shadow Tactics [7]. En el caso de desesperados, se usó el motor gráfico Vision [8], un motor ya obsoleto.

Para este desarrollo se usará el motor y editor Unity 3D, los motivos de esta decisión se explicarán en el apartado 4.1 Entorno de desarrollo.

3. Definición del juego

Rogue Drones es un juego de táctica en tiempo real en que tomaremos el control de un IA, la cual ha sido recién creada y solo sabe que tiene como objetivo encontrar una persona que aparece en sus registros como su creador y de la cual espera recibir nuevas sus instrucciones. La única posibilidad de interacción que tiene esta IA con el mundo físico es un pequeño dron con una cámara y un dispositivo de comunicaciones. Está IA inicia su funcionamiento en una casa destruida, en un país sumido en una guerra civil, pero, el dispositivo de comunicaciones de su dron da la posibilidad de capturar los diferentes drones que usan los ejércitos. Podrá usar los distintos drones capturados para abrirse paso por los diferentes ejércitos, cada uno de estos drones tendrá unas propiedades únicas, por lo cual tendremos que ver tipo de dron es más adecuado para usar en cada ocasión. Los drones no son los únicos enemigos, también habrá defensas enemigas a las que enfrentarse y, por si fuera poco, también habrá que gestionar los recursos que utilizan los drones, así como los obstáculos que pondrán los enemigos.

3.1 Historia

Como hemos introducido en apartados anteriores, en Rogue Drones tomaremos el control de un IA, la cual ha sido recién creada y solo sabe que tiene como objetivo encontrar una persona que aparece en sus registros y esperar sus instrucciones.

La historia está ubicada en un país imaginario y en un futuro cercano donde los ejércitos utilizan drones controlados remotamente para combatir. El país se ha visto envuelto en un conflicto civil que engloba a dos facciones rivales.

Durante el TFM no ha habido tiempo a desarrollar una narrativa compleja, pero si hubiese habido más tiempo, el objetivo sería que los objetivos de las facciones se fueran descubriendo por parte del jugador a lo largo del juego, contemplando las acciones efectuadas por ambos ejércitos, sus mensajes, tanto oficiales como de propaganda, informativos, etc. A partir de aquí, y a raíz de lo que descubriese el jugador, este podría involucrarse en el conflicto que asola su país, pero de la forma que él decida.

3.2 Interacción entre los actores del juego

Como se ha explicado en anteriores apartados, tendremos un dron principal que nos permitirá ver una parte del escenario, que estará limitado a lo que este dron principal puede captar, si el usuario sale de los límites estará fuera de la cobertura de la IA y sufrirá daño por salirse de la zona de cobertura.

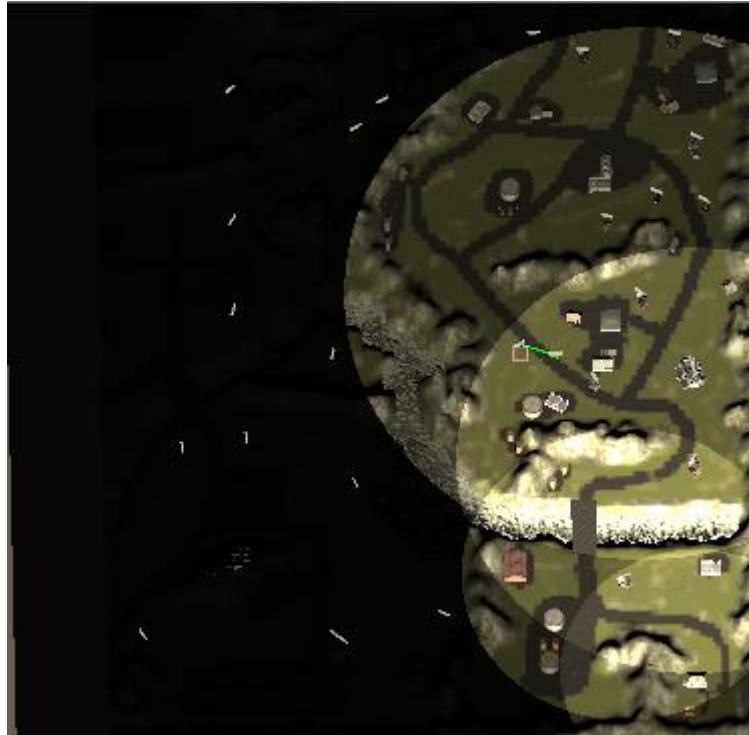


Ilustración 2 Muestra de la zona de cobertura en el escenario de juego.

Lo primero que tendremos que realizar en una misión será capturar drones para poder utilizarlos, esto está limitado a la capacidad de cómputo que se disponga y el coste de computación, lo cual supone una mecánica de gestión de recursos. Si se pierde el dron principal, la partida habrá finalizado.



Ilustración 3 Dron principal capturando otro dron.

Para cumplir los objetivos el jugador, que en este caso será llegar al final del nivel, tendrá que eliminar los enemigos que le entorpezcan la resolución, evitar ser visto o abrir nuevos caminos destruyendo obstáculos como los inhibidores de señal o las armas fijas.

El jugador podrá moverse libremente por el escenario, pero solo podrá ir si tiene la potencia de emisión suficiente para llegar a esa zona. Esta potencia se incrementa con los repetidores que están en algunas de las zonas del juego. Si son capturados, incrementaremos la zona de control de los drones que *hackearemos*.



Ilustración 4 Torre de repetición de señales

Se podrá *hackear* cualquier dron, pero el número está limitado a la capacidad de cómputo que se disponga. Cada dron tiene un coste de cómputo, y podremos tener en nuestro grupo tantos como la capacidad de computación lo permita. La capacidad de cómputo se puede incrementar capturando puestos de mando enemigos.



Ilustración 5 Puesto de mando

Si el dron es destruido (tanto por el enemigo como el propio jugador), su coste de captura volverá a estar disponible para el jugador para su uso.

Cada dron tiene asociado un indicador de estado y un indicador de munición. La salud se puede restablecer en centros de reparación, mientras que la

munición se puede restablecer en polvorines. También existen drones de apoyo que cumplen estas funciones.

Si un dron se queda sin vida, este quedará destruido y no se podrá reparar, por lo que el jugador tendrá que capturar uno de nuevo.

Los drones no serán los únicos enemigos del jugador, también tendremos estructuras como armas defensivas, que no podrán ser pirateadas, o el caso de los inhibidores, que inutilizarán un dron si entra en su zona de alcance. Podemos inutilizar estas estructuras con potencia de fuego o inutilizando los generadores de energía que tienen asociados.

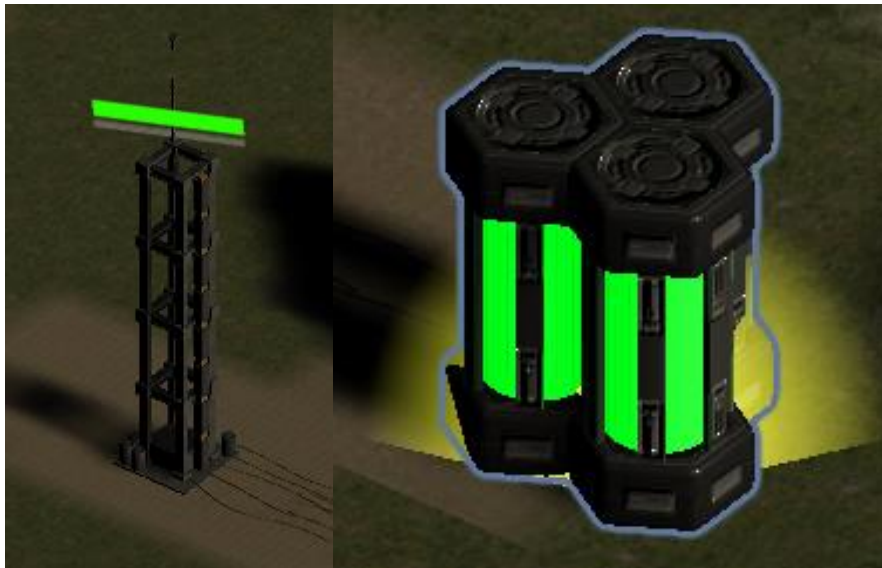




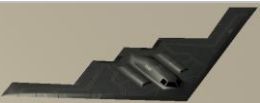
Ilustración 6 Torre de inhibición y su generador.

3.3 Definición de los personajes / elementos


El protagonista de nuestro juego será la IA, a la cual controlará el jugador, y que tiene como objetivo llegar al Creador, la persona que la creó, de la cual ha recibido una solicitud de ayuda. La única forma de llegar a él será un pequeño dron de exploración que tiene la posibilidad de piratear otros drones.

Como se puede deducir de lo indicado anteriormente, nuestro elemento jugable principal serán los drones capturados. Pero por la propia mecánica del juego, estos drones también serán los enemigos del jugador.



El juego dispone de varios tipos de *drones* que podemos controlar y/o enfrentarnos, cada uno con sus propias características, y que además cambian en función del ejército al cual pertenezcan, una versión simplificada sería:

Nombre	Tipo	Descripción	Ataque / Acción	Imagen
Dron principal	Capturador Aéreo	Dron que permite capturar a otros drones e infraestructuras. Si es destruido la partida finalizará.	El tiempo de captura de un dron o estructura depende de la capacidad de cómputo necesaria para la captura. Alcance de ataque corto. Frecuencia de ataque Continua. Ataca a objetivos aéreos y terrestres.	
Caza	Aéreo	Ataca con misiles que persiguen a los objetivos.	Alcance de ataque medio. Daño medio. Frecuencia de ataque media. Ataca a objetivos aéreos y terrestres.	
Bombardero	Aéreo	Ataca con bombas que infligen un gran daño de área.	Alcance de ataque corto. Daño elevado.	


			<p>Frecuencia de ataque lenta.</p> <p>Ataca a objetivos terrestres.</p>	
Explorador	Terrestre	Posee gran velocidad de movimiento, aunque poca resistencia.	<p>Alcance de ataque corto.</p> <p>Daño bajo.</p> <p>Frecuencia de ataque alta.</p> <p>Ataca a objetivos aéreos y terrestres.</p>	
Tanque	Terrestre	Posee poca velocidad de movimiento, aunque tiene una gran resistencia.	<p>Alcance de ataque elevado.</p> <p>Daño elevado.</p> <p>Frecuencia de ataque lenta.</p> <p>Ataca a objetivos terrestres.</p>	
PEM	Operaciones especiales Aéreo	Provoca daño PEM a objetivos terrestres y aéreos, lo que reduce su velocidad y les hace daño mientras estén en la zona de alcance del dron.	<p>Alcance de ataque medio.</p> <p>Daño bajo.</p> <p>Frecuencia de ataque continua.</p> <p>Ataca a objetivos aéreos y terrestres.</p>	
Sanador	Apoyo	Repara los	Alcance de	

	auxiliar Aéreo	drones aliados	sanación baja. Sanación baja. Frecuencia de sanación continua. Sana a objetivos aéreos y terrestres.	
--	-----------------------	----------------	---	---

Aparte de los drones, tendremos diversas infraestructuras, que serán tanto objetivos importantes a capturar como a destruir:

Nombre	Capturable	Descripción	Ataque / Acción	Imagen
Repetidores de señal	Si	Permite ampliar el alcance de las señales enviadas por la IA, lo que a nivel jugable se traduce en las zonas del mapa que el jugador puede recorrer.	Descubre la zona del mapa sobre la que tiene alcance. Si un dron aliado sale de su zona de alcance sufrirá daño PEM (ver siguiente estructura)	
Inhibidores de señal	No	Los drones capturados no pueden pasar por su zona de alcance ya que reciben daño de PEM. Es necesario destruir sus generadores de energía para inutilizarlos.	Baja la velocidad de los drones y reciben daño constante.	

Generadores	No	Generadores de energía que permiten al resto de infraestructuras funcionar (Ver inhibidores de señal).	Si es destruido, los inhibidores de señal quedan inhabilitados	
Torres Antiaéreas	No	Ataca a objetivos aéreos que entren en su zona de alcance con misiles que los perseguirán.	Alcance de ataque medio. Daño medio. Frecuencia de ataque media. Ataca a objetivos aéreos.	
Torres Artilladas	No	Ataca a objetivos terrestres que entren en su zona de alcance.	Alcance de ataque medio. Daño medio. Frecuencia de ataque media. Ataca a objetivos terrestres.	
Puestos de mando móvil	Si	Incrementa la capacidad de cómputo, necesaria para capturar drones más poderosos	Añade una unidad más de cómputo al jugador cuando es capturada.	
Centros de reparación	Si	Repara los drones aliados	Alcance de sanación baja. Sanación baja.	

			<p>Frecuencia de sanación continua.</p> <p>Sana a objetivos aéreos y terrestres.</p>	
Polvorines	Si	<p>Permiten reponer la munición de los drones. Crean una gran explosión si son destruidos</p>	<p>Regenera la munición de los drones capturados por el jugador</p>	

3.4 Objetivos planteados al jugador

Como en el TFM solo se dispone de un tiempo limitado, el objetivo del escenario será simplemente llegar a la ubicación del creador, superando de camino a los ejércitos enemigos. El jugador tendrá que abrirse paso teniendo las limitaciones que tendrá respecto a los drones que podrá capturar respecto al número y poder de las fuerzas enemigas.

4. Diseño técnico

4.1 Entorno de desarrollo

A lo largo de las prácticas efectuadas durante este máster, se ha usado el motor Unity como base para explicar los conceptos de programación de videojuegos, tanto en 2D como 3D, así como conceptos de Inteligencia artificial o juegos multijugador. Con el objetivo de usar los conocimientos aprendidos, se volverá a contar con este motor, en concreto la versión Unity 2019.3.2f1 para desarrollar el proyecto.

Para programar los diferentes scripts que comprenderá el juego, se usará el IDE Visual Studio 2017, que da soporte al motor Unity.

Como gestión de versiones, a petición de los profesores, se usará el repositorio GitHub.

4.2 Requerimientos técnicos

Para el desarrollo de este TFM no se ha utilizado ningún hardware especial para el desarrollo del juego como por ejemplos tabletas de dibujo, pads de control o gafas de realidad virtual. El entorno de desarrollo ha consistido en un pc estándar y puede replicarse en cualquier entorno.

Lo único necesario han sido las herramientas que se mencionan en el siguiente apartado.

4.3 Herramientas empleadas

A continuación se enumeran las diferentes herramientas utilizadas durante el desarrollo de este TFM:

Herramienta	Versión	Uso
Unity 3D	2019.3.2f1	Motor del juego y editor principal de este. Se ha usado como herramienta principal durante su desarrollo.
Visual Studio 2017	15.9.6	IDE usado para desarrollar y compilar las diferentes clases de C# utilizadas por el motor Unity para implementar la lógica del juego.
Git Hub	Herramienta online	Repositorio usado para el control de versiones del juego, copia de seguridad y método de entrega del código a los profesores.
Git Kraken	6.6.0	Herramienta para gestionar las diferentes peticiones para subir y bajar código de Git Hub.
Paint/Paint3D	Incluido en SO Windows	Herramienta para editar imágenes. Usadas para modificar crear algunas de

		las texturas que se usan en el juego.
ProBuilder		Componente de Unity que permite modificar los diferentes elementos 3D usados
Split Unity Terrain Script		Script que incorpora un editor de Unity que permite editar los terrenos para dividirlos en otros más pequeños [11]

4.3 Assets y recursos al juego

La mayoría de los recursos del juego se han obtenido de la Asset Store de Unity, es necesario indicar que todos estos elementos son gratuitos a la fecha del desarrollo de este TFM.

Nombre	Dirección web	Uso en el juego
Low poly combat drone	https://assetstore.unity.com/packages/3d/low-poly-combat-drone-82234	Usado como dron sanador.
Free Pack (Drones)	https://assetstore.unity.com/packages/tools/physics/free-pack-117641	Usado originariamente para el dron principal, finalmente no se usó salvo algunas funcionalidades de su lógica que se adaptaron para animar las aspas del dron principal.
RTS camera	https://assetstore.unity.com/packages/tools/camera/rts-camera-43321	Implementa la funcionalidad de la cámara que se ha usado durante el juego.
Stealth Bomber	https://assetstore.unity.com/packages/3d/vehicles/air/stealth-bomber-56765	Usado como dron bombardero.
Free Sci Fi Gatling Gun - Tower Defense	https://assetstore.unity.com/packages/3d/props/weapons/free-sci-fi-gatling-gun-tower-defense-134222	Usado como torreta artillada.
SciFi Base / Defense Tower - FREE Pack 03	https://assetstore.unity.com/packages/3d/environments/sci-fi/scifi-base-defense-tower-free-pack-03-13295	Usado como torreta antiaérea.
Satellite Dish / Communication Tower	https://assetstore.unity.com/packages/3d/environments/sci-fi/satellite-dish-communication-tower-5884	Usado como repetidor de señal.

HiTech SciFi Energy Cell	https://assetstore.unity.com/packages/3d/environments/sci-fi/hitech-scifi-energy-cell-154526	Usado como generador de energía.
RTS Sci-Fi Game Assets v1	https://assetstore.unity.com/packages/3d/environments/sci-fi/rtscifi-game-assets-v1-112251	Usado como armería, barracón y puesto de mando.
Sci-Fi Environment	https://assetstore.unity.com/packages/3d/environments/sci-fi-environment-84496	Usado como torre PEM y diversos elementos del escenario.
Retro Sci-Fi Pack	https://assetstore.unity.com/packages/audio/sound-fx/retro-sci-fi-pack-61781	Usado para implementar diversos efectos sonoros en el juego.
Ultra Sci-Fi Game Audio UI Pack Vol. 1	https://assetstore.unity.com/packages/audio/sound-fx/ultra-sci-fi-game-audio-ui-pack-vol-1-155417	Usado para implementar diversos efectos sonoros en el juego.
Space Warrior Jet	https://assetstore.unity.com/packages/3d/vehicles/space/space-warrior-jet-23174	Usado como dron caza.
SciFi UI Sound FX	https://assetstore.unity.com/packages/audio/sound-fx/scifi-ui-sound-fx-27282	Usado para implementar diversos efectos sonoros en el juego.
Unity Samples: UI	https://assetstore.unity.com/packages/essentials/unity-samples-ui-25468	Usado para implementar los elementos de la interfaz gráfica del juego, desde los menús hasta el HUD.
Ultra Sci-Fi Game Audio Weapons Pack Vol. 1	https://assetstore.unity.com/packages/audio/sound-fx/weapons/ultra-sci-fi-game-audio-weapons-pack-vol-1-113047	Usado para implementar diversos efectos sonoros en el juego.
Realistic Drone	https://assetstore.unity.com/packages/3d/vehicles/air/realistic-drone-66698	Usado como dron principal.
Free Missile	https://assetstore.unity.com/packages/3d/props/weapons/free-missile-72692	Usado como misiles lanzados por drones y torres antiaéreas.
MSTA Russian Artillery Tank	https://assetstore.unity.com/packages/3d/vehicles/msta-russian-artillery-tank-101487	Usado como dron tanque
Rockets, Missiles &	https://assetstore.unity.com	Usado como bomba

Bombs - Cartoon Low Poly Pack	m/packages/3d/props/weapons/rockets-missiles-bombs-cartoon-low-poly-pack-73141	lanzada por el dron bombardero.
Alien Ships Pack	https://assetstore.unity.com/packages/3d/vehicles/space/alien-ships-pack-131137	Usado como dron PEM.
Project Car - Raider	https://assetstore.unity.com/packages/3d/vehicles/land/project-car-raider-32795	Usado como dron explorador
Cool Visual Effects	https://assetstore.unity.com/packages/vfx/particles/cool-visual-effects-161342	Usado para implementar diversos efectos visuales en el juego.
Effect textures and prefabs	https://assetstore.unity.com/packages/vfx/particles/effect-textures-and-prefabs-109031	Usado para implementar diversos efectos visuales en el juego.
Volumetric Lines	https://assetstore.unity.com/packages/tools/particles-effects/volumetric-lines-29160	Usado para implementar los rayos usados por el dron principal y el dron sanador.
Simple Health Bar FREE	https://assetstore.unity.com/packages/tools/gui/simple-health-bar-free-95420	Usado para implementar las barras de vida y salud de los drones y estructuras.
Abandoned buildings	https://assetstore.unity.com/packages/3d/environments/abandoned-buildings-62875	Usado como elementos del escenario.
Industrial Buildings & Props	https://assetstore.unity.com/packages/3d/environments/industrial/industrial-buildings-props-13173	Usado como elementos del escenario.
Standard Assets	https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2017-3-32351	Usado para generar el escenario del juego
Prototyping Pack (Free)	https://assetstore.unity.com/packages/3d/prototyping-g-pack-free-94277	Usado para generar el escenario del juego
RPG/FPS Game Assets for PC/Mobile (Industrial Set v2.0)	https://assetstore.unity.com/packages/3d/environments/industrial/rpg-fps-game-assets-for-pc-mobile-industrial-set-v2-0-	Usado para generar el escenario del juego

	86679	
Yughues Free Pavement Materials	https://assetstore.unity.com/packages/2d/textures-materials/roads/yughues-free-pavement-materials-12952	Usado para generar el escenario del juego
Russian buildings lowpoly pack	https://assetstore.unity.com/packages/3d/environments/urban/russian-buildings-lowpoly-pack-80518	Usado para generar el escenario del juego
FogOfWar.shader	http://staraban.com	Usado para hacer la niebla la guerra

También es importante añadir que se han usado diversos sonidos provenientes de las siguientes páginas:

- <http://soundbible.com/>
- <https://www.freesoundeffects.com/>
- <https://freesound.org/>

4.4 Esquema de arquitectura del juego / componentes

Como durante el desarrollo de este TFM se han desarrollado múltiples clases que implementan la lógica del juego, se van a agrupar en función del elemento que gestionan:

4.4.1 Drones

4.4.1.1 BasicDrone

Clase que implementa los aspectos básicos del funcionamiento de un dron, desde la gestión de la salud, el proceso de captura por parte del jugador, la munición empleada o la gestión del daño recibido. A continuación, se detalla más en concreto las funcionalidades implementadas:

- **Gestión de daño:** Cuando un dron recibe daño este se resta de su salud total, el proceso es inverso si el dron es sanado por un dron sanador o por un barracón. Si la salud total llega a 0 el dron será destruido. Antes de ser destruido, se mostrarán efectos de humo y fuego cuando la salud del dron baje del 50% y cuando solo le quede un 25% de salud.
- **Gestión de daño PEM:** Si el dron recibe daño de PEM, el dron recibe el daño indicado y su velocidad de movimiento se reduce al mínimo.
- **Gestión de munición:** Si el dron tiene un límite de ataques, este es reducido una vez que se efectúa el ataque. Si se queda sin usos el dron no podrá atacar. El límite de ataques se puede recuperar con la

infraestructura polvorín, que añadirá la munición indicada al límite de ataques hasta llegar a su máximo.

- **Gestión de captura:** Si se ha solicitado la captura de un dron, lo primero que se hará será revisar con la clase `GameplayManager` si con la capacidad de cómputo actual es posible capturar el dron. En caso afirmativo, se irá rellenando el indicador de captura en función del tiempo en que se esté efectuando el proceso de captura y el coste del dron a capturar. Cuando más cueste de capturar el dron, más tiempo tardará en ser capturado. En cuando el dron sea capturado, se deshabilitará su IA y pasará a poder ser controlado por el jugador. En caso de que el dron no sea capturable, no se realizara ningún cambio en el indicador de captura y se emitirá un sonido para indicar al jugador que no se está efectuando la captura.
- **Gestión de cobertura:** Si el dron sale de la cobertura de un repetidor de señal, este repetidor indica al dron de que está fuera de cobertura, por lo que se procede a activar el daño PEM. Esto es desactivado una vez el dron vuelve a la cobertura de un repetidor.

4.4.1.2 DroneInterface

Interfaz que define las funciones que se utilizan para todos los drones del juego. Estas funciones se usan para comunicación entre colliders de los diferentes objetos, definir las funciones de ataque y captura u obtener el área de ataque del dron.

Clases específicas de un dron

Las siguientes clases implementan las funcionalidades específicas de un tipo de dron concreto. También incorporan funcionalidades básicas que si bien pueden ser generales a todos los drones como el ataque o la gestión de los diferentes estados de la IA (consultar su apartado específico para más información), puede haber diferentes variables que obligan a gestionarlas por cada dron de forma individual.

4.4.1.3 AirDrone

Implementa las funcionalidades del drone caza. Este dron tiene la particularidad de que genera un objeto misil cada vez que ataca a un objetivo (terrestre o aéreo).

4.4.1.4 BomberDrone

Implementa las funcionalidades del drone bombardero. Como particularidad este dron genera un objeto bomba cada vez que ataca a un objetivo terrestre.

4.4.1.5 HealerDrone

Implementa las funcionalidades del dron sanador. Este dron no tiene una función de ataque si no de sanación. Para ello enviamos el valor de la sanación efectuada a la clase BasicDrone. Los objetivos de este dron pueden ser tanto drones enemigos como drones capturados. Para reflejar el ataque se genera un objeto lightbeam (ver 4.3 Assets y recursos al juego) que conecta el dron con su objetivo.

4.4.1.6 MainDrone

Implementa las funcionalidades del dron principal. Este dron se distingue de los demás en que es el único que puede capturar los drones enemigos para que pasen a ser controlados por el jugador. El proceso es similar al proceso de ataque, solo que en lugar de enviar los valores de daño al BasicDrone del objetivo, se invoca a su función capture. Al igual que el dron sanador, para reflejar el proceso de captura se genera un objeto lightbeam (ver 4.3 Assets y recursos al juego) que conecta el dron con su objetivo.

4.4.1.7 PEMDrone

Implementa las funcionalidades del dron PEM. Este dron inflige daño PEM a los drones enemigos que entran en su zona de efecto. Para ello detecta que drones entran en su collider y/o se mantiene dentro de él. Una vez detectado se comunica la clase BasicDrone que active el daño PEM, si el dron sale del área de efecto se desactiva este daño. Para reflejar el proceso, se activa un efecto para mostrar que el dron está efectuando el daño.

4.4.1.8 ScoutDrone

Implementa las funcionalidades del dron explorador. Ataca a unidades terrestres y aéreas que entren en su collider siguiendo el mismo proceso de ataque que se indica en los drones anteriormente explicados. Para realizar el ataque se genera un efecto que simula el disparo. Es importante indicar que en el proceso de patrulla se ha modificado la velocidad del agente de IA una vez que se acerca a un waypoint.

4.4.1.9 TankDrone

Implementa las funcionalidades del dron tanque. Su funcionamiento es similar al del dron explorador, pero limitando el ataque a solo unidades terrestres.

4.4.2 Armas

4.4.2.1 Bomb

Implementa las funcionalidades de las bombas lanzadas por el dron bombardero. Una vez la bomba colisiona con el escenario o con el objetivo, se recoge todos los objetos enemigos que están en el área de efecto (collider de la bomba) y se le inflige el daño designado a la bomba

4.4.2.2 Missile

Implementa el comportamiento de los misiles lanzados por las torres antiaéreas y los drones caza.

Usando los parámetros introducidos de velocidad, seguirá al objetivo asignado durante 5 segundos, si en ese tiempo no se ha colisionado con el enemigo el misil se autodestruirá.

4.4.2.3 Missile_Collider

Comunica al script Missile que ha colisionado con el objetivo.

4.4.2.4 Missile_Launcher

Implementa el lanzamiento de un misil. Para ello instancia un nuevo objeto misil con los datos asignados (velocidad, daño y objetivo a perseguir). Esto lo realiza continuamente en un intervalo de tiempo asignado por el programador.

4.4.3 Estructuras

4.4.3.1 BasicStructure

El funcionamiento es similar al script BasicDrone pero adaptándolo a las estructuras del juego, implementando las funcionalidades básicas como la salud, el daño recibido o la captura. A continuación, se detalla más en concreto las funcionalidades implementadas:

- **Gestión de daño:** Cuando una estructura recibe daño este se resta de su salud total, siendo destruida la estructura si llega a 0. A diferencia de los drones, no se destruirá el elemento cuando sea destruido, se quedarán inutilizadas mostrando el efecto de fuego para representar la destrucción de la estructura.
- **Gestión de captura:** A diferencia de los drones no es necesario comprobar la capacidad de cómputo porque no hay coste al capturar una estructura, pero si se comprobara si la estructura es capturable por el jugador o no. En caso afirmativo, el proceso de captura será idéntico al explicado en el script BasicDrone.

4.4.3.2 StructuresInterfaces

Interfaz que define las funciones que se utilizan para todas las estructuras del juego como la comunicación entre colliders de los diferentes objetos o definir las funciones de ataque y captura.

Clases específicas de una estructura

Al igual que con la implementación de los drones, las siguientes clases implementan las funcionalidades específicas de un tipo de estructura específica. Incorporan funcionalidades básicas como el ataque o la acción específica de cada estructura.

4.4.3.4 AA_Gun

Implementa el funcionamiento de las torres antiaéreas. Si un dron aliado de tipo aéreo entra en su zona de efecto, se instancia un elemento misil que perseguirá al objetivo durante unos segundos. Ira disparando un nuevo misil cada vez que pase el tiempo asignado entre disparos.

4.4.3.5 AA_Rotation

Implementa la rotación de las torres antiaéreas. Simplemente asigna una fuerza a la rotación de la torre antiaérea para que rote sobre sí misma.

4.4.3.6 Armory

Implementa el funcionamiento de los polvorines. Si un dron aliado se acerca a su zona de efecto, se le incrementara la munición. Para ello, una vez el dron contacta con su collider, se comprueba que el dron usa munición, y si la usa se comprueba que no está a su nivel máximo. Si no lo está se incrementará su número hasta alcanzar su máximo, mediante llamadas al script BasicDrone. Mientras se esté recargando la munición de un dron se mostrará un efecto y sonido para demostrar que la estructura está en funcionamiento. En caso de que esta estructura sea destruida, se creará un gran daño de área que afectará a todos los drones (aliados o no) que estén en el área de efecto de la estructura.

4.4.3.7 Barracs

Implementa el funcionamiento de los centros de reparación. Su funcionamiento es idéntico al del script Armory, pero sustituyendo la recarga de munición por la recarga de vida. A diferencia del Armory, no genera daño de área cuando la estructura es destruida.

4.4.3.8 CommandPost

Implementa el funcionamiento de los puestos de mando móvil. Si el jugador captura esta estructura, se incrementa la capacidad de cómputo del jugador. En este proceso se guardará la estructura en la lista de elementos capturados temporales (ubicada en el script GameConstant) para que permanezca como tal hasta que el jugador capture una torre de repetición, manteniendo, a partir de ese momento, el puesto capturado si se reinicia el nivel, además de comunicarlo al script GameplayManager, con el objetivo que otros elementos

sepan la capacidad de cómputo del jugador y que también se muestra en el HUD.

4.4.3.9 GatlingGunCustom

Implementa el funcionamiento de las torres artilladas. Es una modificación del script GatlingGun del asset Free Sci Fi Gatling Gun - Tower Defense. Incorpora la funcionalidad de atacar drones aliados terrestres. El método es idéntico al del script AA_Gun, sustituyendo los misiles por un efecto (y sonido) de un disparo.

4.4.3.10 Generator

Implementa el funcionamiento de los generadores de las torres PEM. Su funcionamiento estándar se controla por el script BasicStructure, pero este script implementa una funcionalidad similar a la vista en el script CommandPost. Si el generador es destruido se guardará la estructura en la lista de elementos capturados temporales para que permanezca hasta que el jugador capture una torre de repetición, manteniendo, a partir de ese momento, el generador destruido (y la torre PEM asociada inhabilitada) si se reinicia el nivel.

4.4.3.11 PEMTower

Implementa el funcionamiento de los inhibidores de señal. Su funcionamiento es similar al del script PEMDrone. En este caso solo se atacará con ataque PEM a los drones aliados que entren en su zona de efecto. Esta estructura no es capturable, por lo que la única forma de parar su ataque es destruyendo la estructura o destruyendo el generador asociado a esta torre. Si el script detecta que el objeto generador ha sido destruido, inhabilitará el ataque de la estructura.

4.4.3.12 Radar

Implementa el funcionamiento de los repetidores de señal. Si la estructura es capturada, se comunicará al script FogOfWar la posición y rango de cobertura del repetidor para que desvele la zona de cobertura en la niebla de guerra. Si un dron aliado no está en la zona de efecto del repetidor (implementado con un collider), recibirá daño por PEM, hasta que entre de nuevo en la zona de cobertura de un repetidor. En cuando es capturado, se designa como punto de reinicio si el jugador muere. Además, los puestos de mandos capturados y los generadores destruidos se guardarán como tales por si es necesario reiniciar el nivel.

4.4.4 Jugabilidad

4.4.4.1 AuxiliarOperation

Incorpora operaciones que son usadas por todo tipo de scripts a lo largo de todo el proyecto, un listado de estas operaciones es:

- Indicar si un elemento está destruido.
- Indicar si un elemento está capturado
- Indicar si un elemento es del jugador o del enemigo.
- Indicar si un elemento es un dron del jugador o del enemigo.
- Indicar que aliados tiene un elemento.
- Indicar si un elemento es terrestre aéreo.
- Obtener un elemento en la jerarquía del objeto de juego.

4.4.4.2 ColliderCommunicator

Comunica que un objeto a colisionado con su objeto padre dentro de la jerarquía de Unity.

4.4.4.3 CommonInterface

Interfaz común que comparten tanto los objetos de tipo dron como los de tipo estructura. Las funciones que define son las siguientes:

- isDestroyed: indica si el elemento está destruido.
- isCaptured: indica si el elemento está capturado por el jugador.
- isAerial: indica si el elemento es aéreo

4.4.4.4 GameConstant

Define las constantes que son usadas a lo largo del proyecto. También define las variables que son usadas entre los diferentes reinicios de niveles cuando el jugador muere como por ejemplo los edificios capturados (Repetidores de señal, generadores y puestos de mando).

4.4.4.5 EndLevel

Implementa el comportamiento a efectuar cuando el jugador llega a la zona de fin de nivel. Es lanzado cuando el jugador contacta con el collider que tiene el objeto del juego asignado, en ese entonces, se cambia al nivel siguiente o a la pantalla de fin del juego. También se reinicia las variables globales que se usan entre reinicio de niveles que están definidas en la clase GameConstant.

4.4.4.6 FogOfWar

Gestiona la revelación de las zonas de la niebla de guerra cuando un repetidor de señal es capturado por el jugador. Se le indica el epicentro y el radio al shader FogOfWar. También se reinicia los valores de este shader cuando se empieza o reinicia un nivel.

4.4.4.7 GameplayManager

Gestiona el fin de la partida, el reinicio del nivel, la activación del menú de pausa y sirve de comunicación entre los objetos del juego y el HUD del jugador. Además, gestiona los drones que han sido capturados por el jugador y la capacidad de cómputo que tiene este para capturar otros drones.

Cada vez que un elemento de juego necesita comunicar algo al HUD como su vida, munición, nivel de CPU actual o su eliminación lo comunicará al GameplayManager y este lo comunicará al HUDManager script. También detectará algunos de estos cambios como cuando el dron principal es eliminado, iniciando o finalizando la partida.

4.4.4.8 Tutorial

Muestra un mensaje de tutorial cuando el jugador alcanza la zona asignada. Es activado cuando el jugador contacta con el collider que tiene el objeto del juego asignado. Cuando esto ocurre, se envía al script TutorialScript el texto y la imagen a mostrar por el tutorial.

4.4.4.9 ApplicationManager

Gestiona el reinicio de los niveles y salida del juego que son llamados por los menús de la interfaz gráfica. Está creado a partir de un script del asset Unity Samples: UI y está ubicado en la carpeta de este.

4.4.5 Jugador

4.4.5.1 PlayerMovement

Gestiona el input del jugador. A continuación, se listan las posibles acciones del jugador y como se tratan en este script:

- Clic izquierdo en una zona del mapa: Si se hace clic izquierdo en una zona del mapa, la posición clicada pasará a ser el destino del agente de IA del dron, moviéndolo a ese lugar o la zona más próxima posible.
- Clic izquierdo en un dron / estructura: en este caso se procederá a seleccionar la unidad: para ellos se activará el elemento selección que tiene todos los objetos dron y estructura. Este elemento es una imagen

de un círculo en un fondo transparente. Si es un elemento aliado el círculo será de color azul y si es enemigo será de color rojo.

- Clic derecho en un dron capturado: El dron clicado pasare a ser el dron controlado por el jugador, siendo este el que reciba el input del jugador.
- Clic derecho en un dron / estructura enemiga: El elemento clicado pasará a ser el objetivo del dron clicado. Para ello se envía el elemento a la función ataque del dron activo.

4.4.6 Interfaz gráfica

4.4.6.1 HUDManager

Dibuja los elementos del HUD, tanto los drones como el nivel de CPU actual. Con los datos enviados por el script GameplayManager, se activará o desactivará los elementos visuales de los drones, mediante el script HUDPlayerDrone, y el nivel de CPU que puede usar el jugador. Cuando uno de estos elementos cambie de cantidad, se redibujará el HUD para mostrar el cambio.

4.4.6.2 HUDPlayerDrone

Dibuja y gestiona la representación de un dron en el HUD. Muestra una imagen de este, su vida actual y si es necesario la munición actual. Se actualiza en cada momento con la instancia del dron. También implementa el botón de destrucción del dron (salvo en el dron principal), que comunicará al script BasiDrone del dron que debe autodestruirse.

4.4.6.3 TutorialScript

Activa el elemento de HUD que define visualmente el tutorial. Se le indica el texto e imagen a mostrar, y al cabo de 5 segundos, se desactiva el tutorial.

4.4.7 Shaders

4.4.7.1 FogOfWar

Este shader consiste en cambiar el alfa (transparencia) del color de una zona concreta del shader. Para ello usamos el epicentro de la zona a mostrar y el radio de esta. Con se comprueba la distancia de cada vértice a esta zona. En caso de que se encuentre en ella se cambia el alfa para que sea transparente. Está inspirado en un shader encontrado en video tutorial [15]

4.4.8 Cámara

4.4.8.1 RTSCam

Este script viene definido por el asset RTS camera (y está ubicado en su misma carpeta). Se ha modificado para que cuando se rote la cámara se ajuste su posición respecto al jugador, ya que en ocasiones este no se mostraba al rotar la cámara. Para ello se han usado métodos y variables que ya estaban definidas en el script.

4.4.9 Escenarios de test

Para el desarrollo de la primera entrega de este TFM, se decidió probar cada uno de los elementos que componen el juego de forma individual, por lo que se hicieron una serie de escenarios para probar cada uno de estos elementos de forma individual. Estos escenarios no forman parte del juego, solo se hicieron como ayuda para validar el desarrollo efectuado y solo son accesibles mediante el editor de Unity.

No estaba contemplado volver a utilizarlos al tener ya un escenario final durante las siguientes entregas, pero se volvieron a utilizar de forma puntual alguno de ellos. Hay que destacar que al añadir más elementos al juego como el HUD y la gestión de recursos, hicieron que algunos de ellos se quedaran obsoletos al necesitar más elementos extra para funcionar correctamente, aun así, pueden servir para validar elementos individuales si se omiten los mensajes de error que provocan estos elementos extra.

Los escenarios desarrollados son los siguientes:

- AA_Test: Test para comprobar el funcionamiento de las torres antiaéreas.
- AirDrone_test: Escenario para probar el dron caza.
- Armory_test: Validación de los polvorines.
- Barrac_test: Prueba de los centros de reparación.
- BombarderDrone_test: Prueba de funcionamiento del dron bombardero.
- CommandPost_test: Validación de los puestos de mando móvil.
- Demo: Escenario que se entregó como juego en la primera entrega. Sirvió como demostración de los elementos y mecánicas desarrolladas hasta entonces.
- Drone_test: Escenario para probar el dron principal.
- HealDrone_test: Prueba de funcionamiento del dron sanador.
- IA_test: Test para comprobar el funcionamiento de la inteligencia artificial.
- Missile_test: Prueba de misiles lanzados por drones y estructuras.
- PEMDrone_test: Escenario para probar el dron PEM.
- PEMTower_test: Test para comprobar el funcionamiento de las torres de inhibición de señal.
- Radar_test: Test para comprobar el funcionamiento de las torres de repetición de señal.

- Scout_test: Escenario para probar el dron explorador.
- Selection_test: Validación de la seleccion de unidades en el HUD.
- TankDrone_test: Prueba del dron tanque.
- Turret_test: Test para comprobar el funcionamiento de las torres artilladas.

4.5 IA de los enemigos

Actualmente hay dos elementos que implementan una inteligencia artificial:

4.5.1 Estructuras

Las estructuras defensivas tienen una IA bastante simple, en la cual si un dron entra en su zona de efecto comienza a dispararle, sanarle o proporcionarle munición hasta que el dron sea destruido o salga de su zona de efecto. En caso de que haya varios objetivos, se centrarán en el que tengan más próximo. El área de efecto de la estructura viene definida por el collider esférico que tienen asociadas las estructuras.



Ilustración 7 Collider de una torre artillada.

4.5.2 Drones

Para la navegación por el escenario, tanto los drones aliados como enemigos usan un nav mesh definido en los escenarios. Este es usado por los nav mesh agent que tienen asignados cada uno de los drones. Hay que añadir que los drones aéreos tienen la posibilidad de saltar ríos y obstáculos usando la opción off mesh path de sus agentes de navegación.

Los drones tienen implementada una máquina de estado que va variando en función de la situación en la que encuentren:

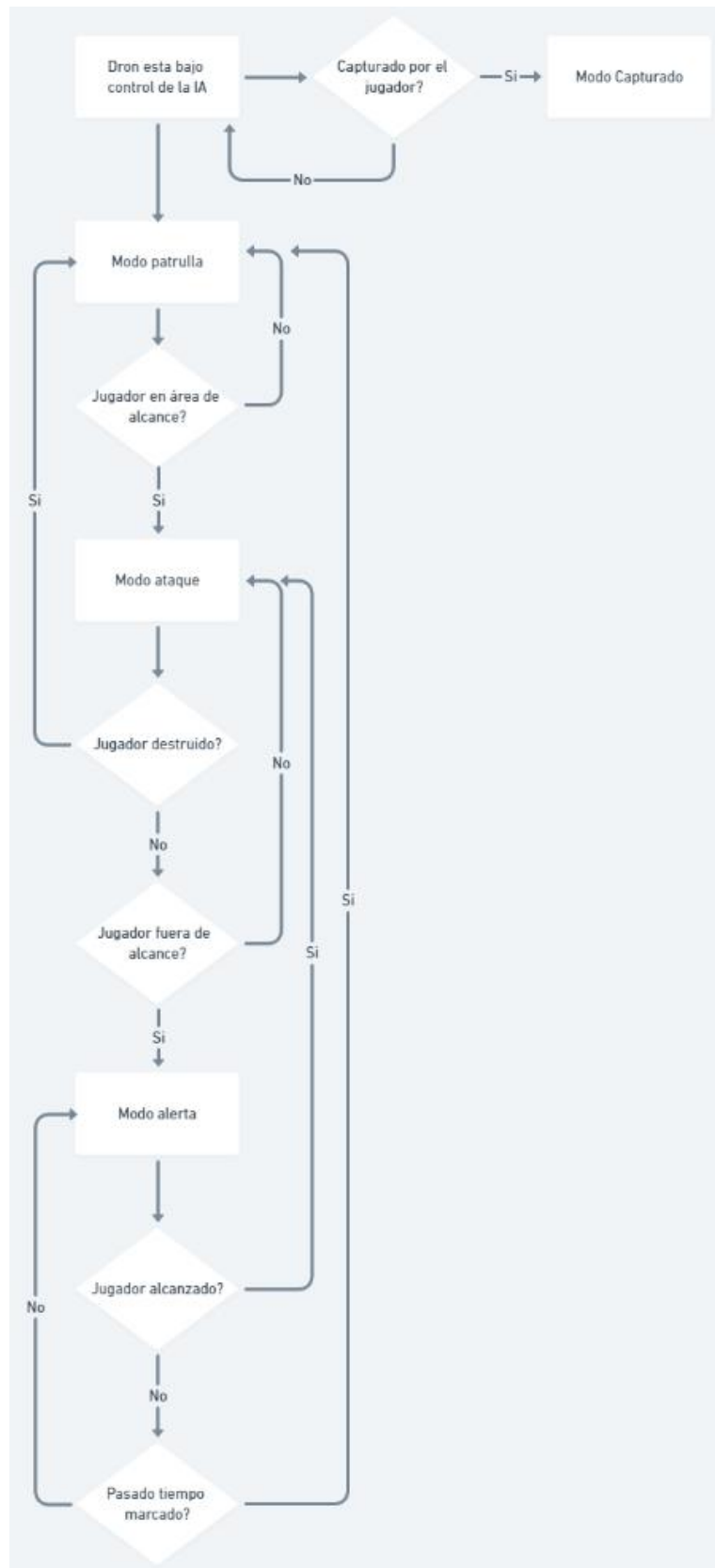


Ilustración 8 Diagrama de flujo de los estados de la IA

4.5.2.1 Patrulla

Todos los drones controlados por la IA disponen de una serie de puntos de ruta (Waypoints) que tienen que seguir. El funcionamiento es el siguiente: el agente de navegación (Nav Mesh Agent) dirigirá el dron al primer punto de ruta de la lista que tenga asociado el dron. Una vez que la distancia entre el dron y el punto de ruta sea mínima, se cambiará el destino del agente al siguiente punto de ruta. Este proceso se repetirá hasta completar la lista de puntos de ruta, momento en que se repetirá el proceso empezando desde el primer punto de ruta de la lista.

Si un dron aliado entra en su zona de alcance, la IA pasará al estado de ataque.

4.5.2.2 Ataque

Todos los drones (al igual que las estructuras) tienen asociado un collider esférico que representa su zona de alcance. Si un dron aliado entra en esta zona, el dron marcará al dron aliado como objetivo y comenzará a atacarlo, hasta que salga de su zona de alcance o lo destruya.

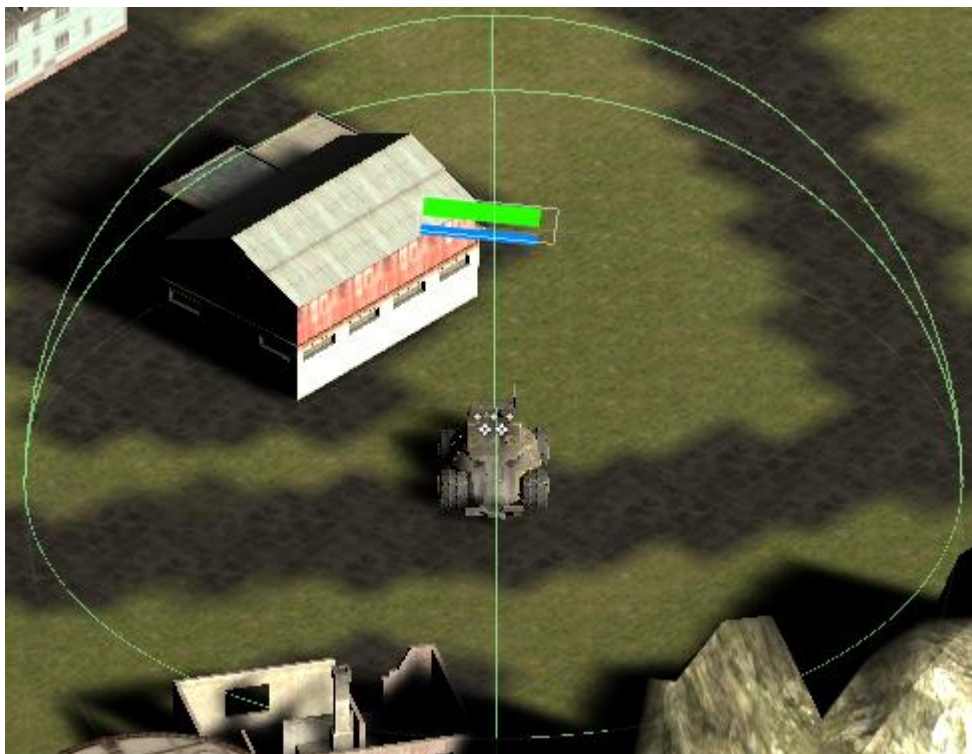


Ilustración 9 Collider de un dron explorador

En caso de que su objetivo salga de la zona de alcance, el dron pasará al estado de la IA de Alerta, Si hay varios objetivos en la zona, los drones se centrarán en el que tenga más cerca.

4.5.2.3 Alerta

Si un dron aliado sale de la zona de efecto/ataque de un dron, este empezará a perseguirte hasta que vuelva a su zona de alcance (o entre otro objetivo distinto) o bien haya pasado el tiempo de alerta estipulado (actualmente 5 segundos).

El proceso es simple, una vez el dron aliado ha salido del collider del dron, se marcará al agente de la IA que mueva el dron a la posición de su objetivo. En caso de no encontrar su objetivo, u otro dron más próximo, en el periodo de tiempo estipulado, el dron pasará al estado de IA de patrulla.

4.5.2.4 Capturado

Cuando un dron es capturado por el jugador, la IA pasa al estado capturado. Este estado hace que la IA no tenga control sobre el dron, haciendo que todas las acciones sean controladas por el jugador.

5. Diseño de niveles

En el juego hay actualmente diseñados 2 niveles. Aunque en ambos la jugabilidad es la misma, tienen objetivos diferentes.

5.1 Tutorial

El objetivo de este nivel es explicar al jugador el funcionamiento del juego, así como las diferentes características de los elementos que lo componen.

Básicamente, el nivel está definido en varias subzonas que tiene como objetivo explicar al jugador el funcionamiento de una mecánica o un dron/estructura concreta.

Estas zonas están todas cohesionadas en el escenario, por lo que para el jugador todo es un mismo nivel.

A continuación, se puede ver una imagen que marca estas zonas y una descripción de ellas:



Ilustración 10 Vista aérea del nivel de tutorial

1. Punto de partida: El jugador empieza en esta zona y se le indica cómo moverse y capturar elementos. El primer elemento es una torre de repetición, explicando de paso el funcionamiento de la cobertura.
2. Introducción del dron explorador. Se le explica al jugador el modo de ataque del dron y podrá capturarlo si quiere.
3. Ubicación del centro de reparación. El jugador no solo aprenderá su funcionamiento, si no que puede recuperar salud de enfrentamientos anteriores y posteriores.
4. Introducción del dron caza. Si el jugador no ha ido hacia la zona 5, no tendrá suficiente CPU para capturarlo, aprendiendo así del límite de cómputo y su impacto en la partida. En ambos casos podrá ver el ataque del dron.
5. Introducción de las torres artilladas. Se le explica al jugador el funcionamiento de estas estructuras y harán daño a los drones exploradores capturados que se acerquen a esta zona, enseñando al jugador su efecto sobre los drones terrestres.
6. Ubicación de puesto de mando móvil. Se le explica su funcionamiento al jugador y le proporciona, una vez capturado, suficiente cantidad de cómputo para capturar todos los tipos de drones, pero no para capturar los drones restantes del escenario. El jugador tendrá que decidir qué drones se queda si llega a un punto que no tiene más capacidad de captura, situación que se dará continuamente en el nivel del juego.
7. Introducción de las torres antiaéreas. La intención es idéntica a la de la zona 5, pero con las torres antiaéreas, atacando al dron principal o al dron caza si se ha capturado. Para abrir un camino se debe atacar con un dron explorador. Si se ha perdido el dron de la zona 2, se dispone de otro en la zona 11.
8. Ubicación del polvorín. Al igual que el centro de reparación, el jugador no solo aprenderá su funcionamiento, sino que puede recuperar munición de enfrentamientos anteriores y posteriores.
9. Introducción del dron PEM. Se le explica al jugador el modo de ataque del dron y puede capturarlo si quiere.
10. Introducción del dron sanador. Se le explica al jugador el modo de ataque del dron y puede capturarlo si quiere. Se puede usar como sustituto del centro de reparación, que está mucho más alejado en el mapa.
11. Introducción del dron tanque. Se le explica al jugador el modo de ataque del dron. En esta zona hay una línea defensiva con torres artilladas y antiaéreas, pensada a para ser destruida usando este dron gracias a su largo alcance.
12. Introducción del dron bombardero. Se le explicará el modo de ataque del dron, el cual está pensado para superar la siguiente zona
13. Introducción de la torre de inhibición de señales. Esta zona está pensada para que los drones no puedan pasar a no ser que un dron aéreo destruya el generador de esta torre. Puede ser sorteada por el dron principal si se dispone de suficiente vida.
14. Fin de nivel. Si un dron entra en esta zona se pasa al nivel de juego.
15. Dron explorador extra. Se ubica en una zona que no es molesta para el jugador por si necesita usarlo.

Para que sirva de guía de este escenario, se ha realizado un video de gameplay, el cual puede ser consultado en el apartado 1.5 Breve resumen de productos obtenidos

5.2 Nivel de juego

Este nivel sería la implementación final de la jugabilidad desarrollada. El jugador debe aplicar lo aprendido en el anterior nivel para superarlo, ya que, en este escenario, también se han definido varias zonas, pero cada una de ellas está pensada para proponer un reto al jugador.

A continuación, se pueden ver estas zonas y una descripción de ellas:



Ilustración 11 Vista aérea del nivel del juego

1. Punto de partida: el jugador empieza en esta zona. Existen torres antiaéreas que impiden que el jugador ataje a la zona 2. Esta función también la hace un dron explorador que es fácilmente esquivable. En esta zona se encuentra el primer repetidor que nos permite avanzar pasada esta zona. Si el jugador observa el mapa podrá encontrar fácilmente un puesto de mando móvil que le permitirá capturar futuros drones.
2. En esta zona encontramos torres artilladas y antiaéreas que dificultaran el paso. Entre ellas se encuentra un dron tanque que podemos usar para destruirlas. También hay una estación de reparación disponible. De camino a la zona 3 hay un dron PEM para dificultar el camino.
3. En esta zona tendremos el generador de la torre de inhibición de la zona 5. También hay un dron sanador y un puesto de mando, pero estarán originariamente fuera de la zona de cobertura. Es posible capturarlos si el jugador se arriesga a recibir daño de PEM. También hay disponible un polvorín disponible para capturar.
4. Segunda torre de repetición de señales si el jugador se aventura a la zona 18 sufrirá daño antiaéreo y de la torre de inhibición que encontrará en el camino, obligándole a ir por la zona 5
5. Se encuentra una torre de inhibición de señales, que el jugador podrá haber deshabilitado en la zona 3. La zona está cubierta por un dron explorador y un dron caza.
6. Esta zona cubre la entrada a la segunda zona del mapa con torres artilladas y un dron tanque que patrulla la zona. También hay torres antiaéreas que impiden que el jugador ataje a la zona 8. Estas torres pueden ser esquivadas o destruidas.
7. En esta zona entraremos a la zona urbana. Tendremos la tercera torre de repetición de señales. La zona está cubierta por un dron bombardero, un dron sanador y un dron PEM. De aquí el jugador podrá avanzar a las zonas 8, 9 ,12
8. Esta zona dispone de un centro de reparación fuertemente cubierto por torres artilladas y antiaéreas. También está cubierto por un dron explorador y el dron PEM que ya pudimos ver en la zona 7.
9. Esta zona sirve como conector con la zona 10. Esta zona está cubierta por un dron caza, un dron explorador y una torre antiaérea. Necesitaremos un dron aéreo para acceder a la zona 10. También se encuentra un puesto de mando.
10. En esta zona se encuentra la última torre de repetición. Está cubierta por un dron caza y torres antiaérea.
11. Segundo punto de entrada a la zona 10. Está cubierta por torres artilladas y antiaéreas y un dron explorador. También hay un centro de reparación disponible.
12. En este punto se unen los drones de la zona 9 y la zona 13. Hay un polvorín y un puesto de mando.
13. En esta zona está ubicado el final del nivel. No será accesible a no ser que el jugador haya capturado la torre de repetición de la zona 10. Está cubierta por torres artilladas y antiaéreas. Respecto a los drones, estará vigilada por unos drones de tipo bombardero, tanque y PEM.
14. Opcional: centro de reparación extra.

15. Opcional: puesto de mando y atajo a la zona 7 previa destrucción del generador de la torre de inhibición de señales.
16. Opcional: puesto de mando extra.
17. Opcional: Atajo a la zona 13, cubierto por torres artilladas y antiaéreas, drones tipo bombardero y explorador.
18. Opcional: puesto de mando extra.

El objetivo del escenario es llegar al punto marcado en el mapa (zona 13). Una vez alcanzado se habrá finalizado el nivel y con él, el juego.

Al igual que el escenario anterior, se ha realizado un gameplay de guía, el cual puede ser consultado en el apartado 1.5 Breve sumario de productos obtenidos

6. Manual de usuario

6.1 Controles

Los controles para jugar al juego son los siguientes:

- Ratón: mueve el puntero.
- Botón izquierdo del ratón: selecciona un elemento cuando tiene el puntero encima, mueve al dron si no hay nada que seleccionar.
- Botón derecho del ratón: ataca al objetivo que tenga el puntero encima. Si se mantiene pulsado se puede rotar la cámara.
- Rueda ratón: aumenta o disminuye el nivel de zoom.
- Tecla escape: abre el menú de pausa.
- Tecla Z: rota la cámara a la izquierda.
- Tecla X: rota la cámara a la derecha.
- Tecla Q: aumenta el nivel de zoom.
- Tecla E: disminuye el nivel de zoom.

6.2 HUD

Durante el desarrollo del juego, toda la información útil para el jugador se mostrará en el HUD. A continuación, se muestra una explicación de cada uno de sus elementos:

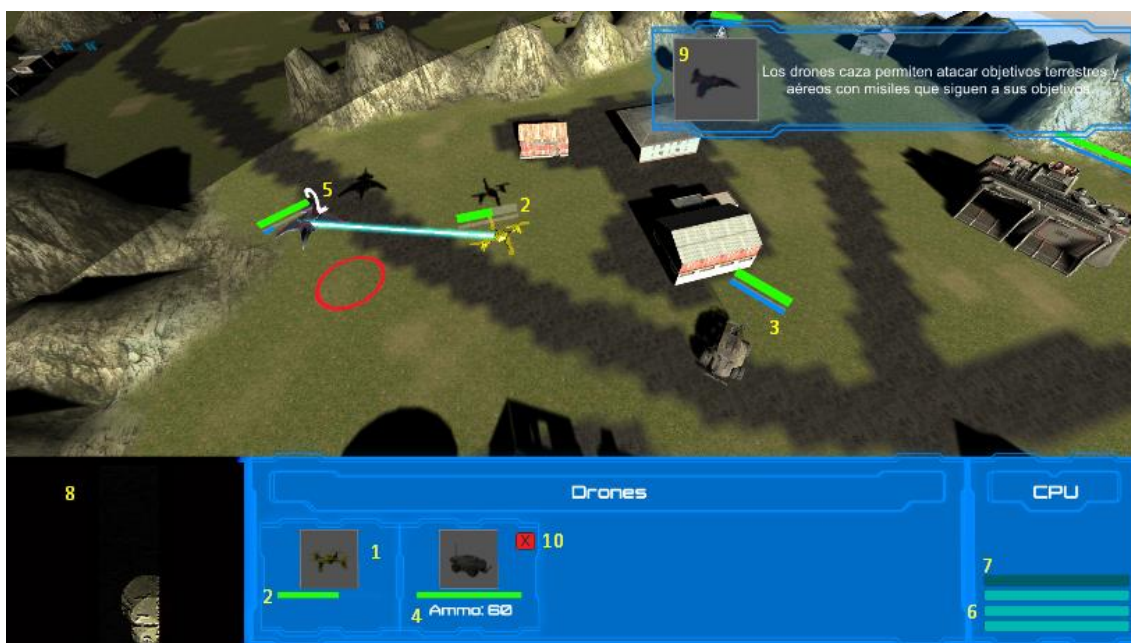


Ilustración 12 Captura del HUD durante la partida.

1. Icono de dron capturado. Si lo pulsamos seleccionaremos el dron y pasaremos a controlarlo (la cámara también se moverá al lugar del dron seleccionado).
2. Barra de vida del dron

3. Barra de captura. Si está completada el dron estará controlado por el jugador.
4. Munición del dron.
5. Coste de captura del dron. Solo aparece en drones enemigos.
6. Unidades de CPU disponibles.
7. Unidades de CPU consumidas.
8. Mini mapa
9. Mensajes de tutorial
10. Botón de destrucción del dron.

6.3 Menús

6.3.1 Menú de inicio

El menú de inicio es la primera pantalla del juego. Desde ella podremos empezar el juego o salir de él.

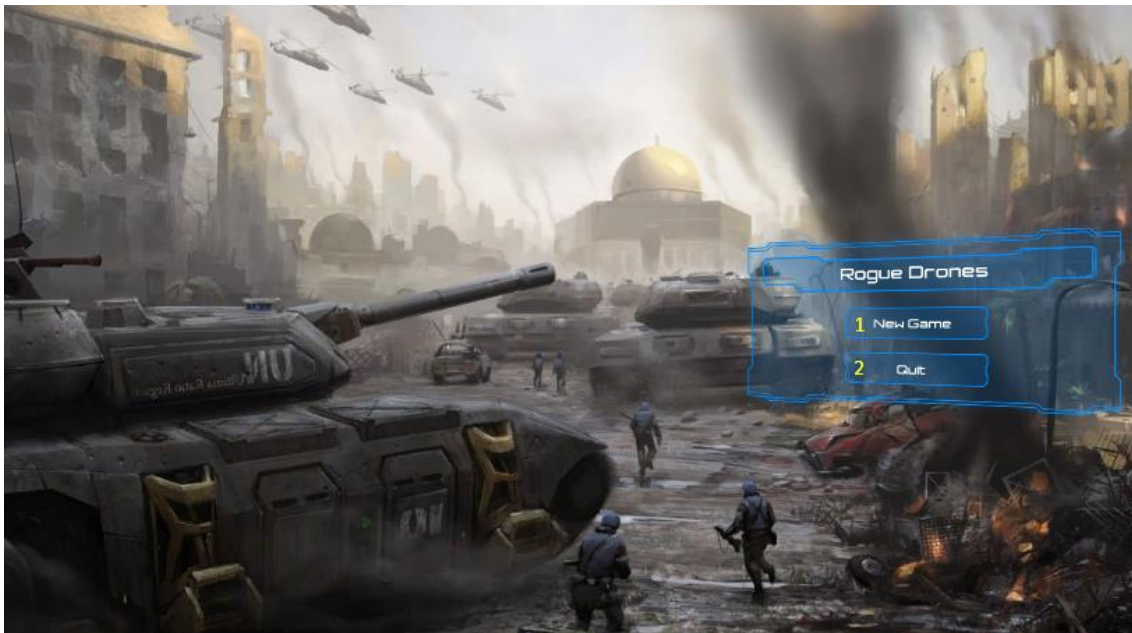


Ilustración 13 Captura del menú de inicio

1. Botón New Game: Inicia el juego desde el primer nivel (tutorial).
2. Botón Quit: Cierra el programa.

6.3.2 Menú de pausa

Al menú de pausa podemos acceder en medio de un nivel jugable pulsando la tecla escape. Desde el podremos reiniciar el juego o salir de él. Es importante indicar que en este proyecto no se ha desarrollado la pausa del juego, por lo que el juego seguirá funcionando cuando se muestre este menú.

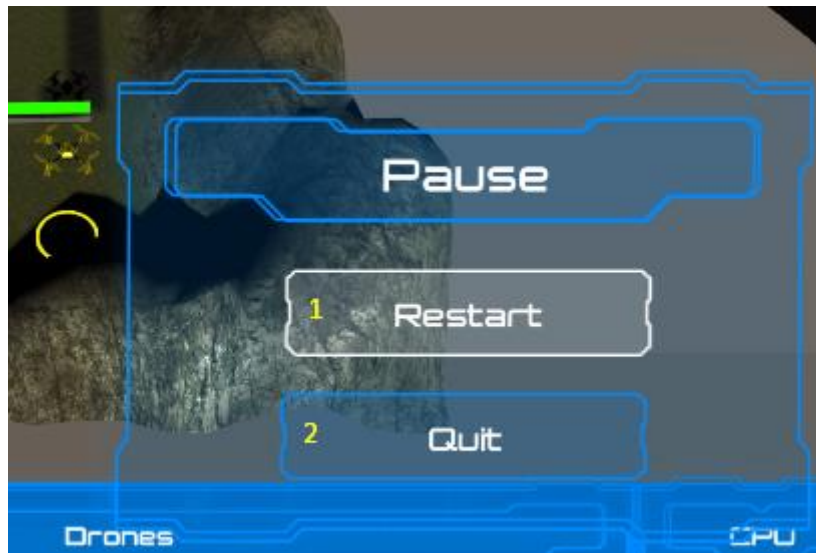


Ilustración 14 Captura del menú de pausa

1. Botón Restart: Reinicia el nivel actual.
2. Botón Quit: Cierra el programa.

6.3.3 Menú de fin del juego

El menú de fin juego aparecerá tanto cuando hayamos superado el segundo nivel como cuando se haya perdido el juego, siendo esta la última pantalla del juego. Desde ella podremos reiniciar el juego el juego o salir de él.



Ilustración 15 Captura del menú de fin del juego

1. Botón Restart: Reinicia el juego desde el primer nivel si se ha completado el juego o desde el último jugado si se ha perdido la partida.
2. Botón Quit: Cierra el programa.

6.4 Desarrollo de la partida

6.4.1 Mover drones

Para mover un dron debemos hacer click izquierdo en la zona a la que lo queramos mover. Si la zona no es accesible se moverá a la zona accesible más cercana.

6.4.2 Captura de drones y estructuras

Teniendo el dron principal seleccionado, hacemos clic derecho sobre el dron o la estructura que queramos capturar. Las estructuras no tienen coste de captura, pero no todas son capturables. Los drones si tienen un coste asociado. Si la captura es posible, la barra de captura del objetivo se incrementará, cuando se complete el dron o la estructura pasará a ser controlados por el jugador. En caso contrario el dron emitirá un sonido indicando que la captura no es posible y la barra de captura no se rellenará.

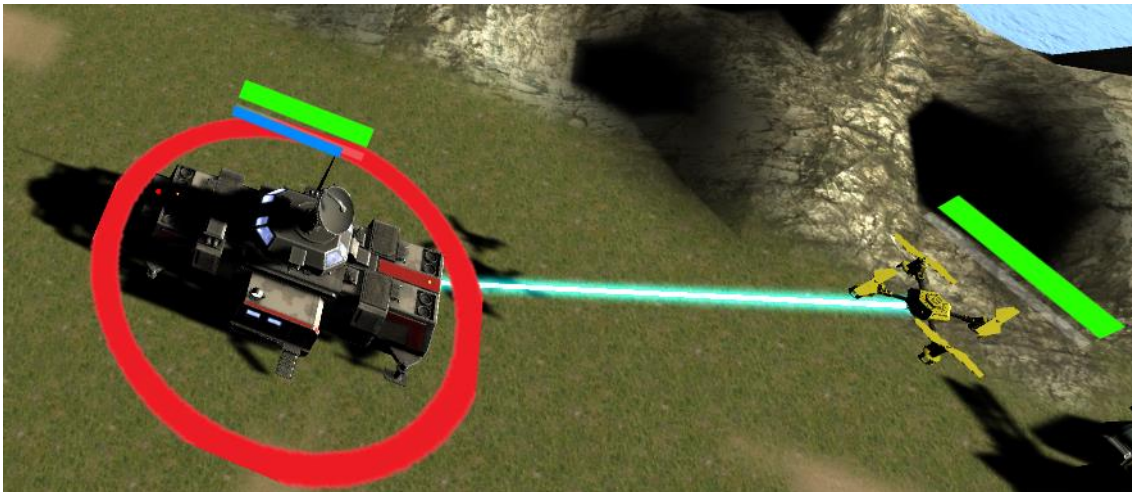


Ilustración 16 Captura de un puesto de mando

6.4.3 Ataque

Para atacar un objetivo, debemos hacer clic derecho con un dron de ataque sobre el que queramos atacar.



Ilustración 17 Dron de ataque atacando a un arma defensiva

6.4.4 Recuperación de salud y munición

Para recuperar salud hay 2 formas posibles:

- Usando un dron sanador: si disponemos de un dron sanador, debemos acercarlo al objetivo de la sanación (o viceversa) para que empiece a sanar el objetivo.
- Usando el centro de reparación capturado: si un dron se acerca a un centro de reparación capturado se irá regenerando su salud mientras esté cerca.



Ilustración 18 Centro de reparación reparando el dron principal

Para recuperar la munición de un dron solo tendremos la posibilidad de acercarlo a un polvorín capturado. Mientras el dron está cerca del polvorín irá recuperando su munición.

6.4.5 Ganar la partida

Para ganar la partida debemos llegar al final del nivel. En el primer nivel simplemente debemos seguir los caminos que se nos abren capturando los diferentes radares. En el segundo nivel debemos llegar a la zona que nos parece descubierta en el mapa desde el inicio de nivel.



Ilustración 19 imágenes de las zonas marcadas como fin de nivel

6.4.6 Fin de la partida

Si nuestro dron principal es destruido, la partida acabara.

Si hemos capturado algún repetidor de señal, aparecemos delante del último que hayamos capturado, manteniendo los repetidores de señal y puestos de mando móvil capturados.

Si no hemos llegado a capturar ninguna aparecerá la pantalla de fin del juego.

6.5 Requisitos mínimos

El juego no requiere de un ordenador potente para funcionar, ya que no se hace uso de una gran potencia grafica ni de cómputo para hacerlo funcional.

Para controlar el juego solo es necesario el ratón, aunque se puede usar el teclado también para controlar aspectos de la cámara.

Solo son necesarios 700 Mb de espacio en disco para ejecutar el ejecutable del juego.

7. Conclusiones

Como conclusión al trabajo realizado durante este proyecto, podemos decir que el objetivo de crear un juego jugable se ha llegado a conseguir. A lo largo del desarrollo del proyecto, el alumno ha tenido que repasar los conocimientos que ha ido adquiriendo durante el transcurso del master para poder desarrollar el entregable con la calidad deseada.

7.1 Lecciones aprendidas

Se repasamos las lecciones que se han aprendido durante el desarrollo de este proyecto, habría que destacar:

- La necesidad de planificar un proyecto complejo de antemano para evitar futuras complicaciones durante el desarrollo
- Definir las mecánicas jugables más básicas primero, y a partir de estas definiciones ampliar a otras mecánicas más complejas.
- Si nuestras mecánicas jugables necesitan de varios actores para funcionar (jugadores, enemigos, infraestructuras...), definirlos de manera que ambos cumplan una función específica y diferente de los demás actores.
- Ajustar las expectativas del desarrollo al tiempo que se dispone para efectuarlo, eliminando mecánicas u objetivos que puedan parecer superfluos, para entregar en proyecto que represente la idea base que se pretende exponer.
- Rebajar las expectativas a los recursos que se encuentren disponibles, por ejemplo, los assets de desarrollo que podamos obtener.
- Aprender cuando es óptimo centrarse en encontrar una solución a un problema o buscar una alternativa a la situación.
- Necesidad de buscar soluciones a los problemas surgidos por diferentes vías (documentación, foros, videos, consultas al profesorado).
- Importancia de planificar con antelación posibles errores y problemas en la planificación y el desarrollo y preparar una contingencia adecuada.

7.2 Reflexión sobre los objetivos planteados

Como el objetivo de realizar un juego jugable implica una definición demasiado amplia, lo vamos a separar en ellos objetivos definidos a realizar durante el desarrollo de las diferentes entregas, de esa forma se podrá abordar cada uno de los aspectos del juego de forma individual:

7.2.1 Inteligencia Artificial

Se ha conseguido definir una IA definida por una máquina de estados básica (más información en el apartado 4.5 IA de los enemigos). El alumno reconoce que, de cara al código, que la implementación de esta podía estar implementada de una forma más encapsulada, pero la necesidad de que el enemigo pase a ser también un elemento jugable dificultó esta opción. Sobre la navegación de los diferentes drones se ha conseguido implementar de una forma óptima, pero el alumno reconoce que respecto a los drones aéreos se podría haber efectuado de una forma que la navegación de estos drones fuese más natural, pero la falta de tiempo dificultó investigar e implementar otras alternativas

7.2.2 Drones

Respecto a los drones, se ha conseguido implementar un dron de cada uno de los tipos que se habían definido en la jugabilidad (3.3 Definición de los personajes / elementos). Cada uno de ellos dispone de un ataque y movilidad característica. También se les han añadido efectos y sonidos. Como aspecto negativo hay que destacar que ha sido difícil implementarles algún tipo de animación, principalmente por limitaciones de tiempo y la necesidad, en algunos casos, de redefinir el asset original. También, a nivel estético, al provenir de diferentes assets, y limitado por la gratuidad de estos, ha sido difícil darles una cohesión estética.

7.2.3 Infraestructuras

El resultado del desarrollo de estos elementos ha sido similar al de los drones. Se les han añadido funcionalidades específicas, efectos, sonido y animaciones en algunos casos. Se ha de destacar el mismo aspecto negativo sobre la estética que también tuvieron los drones.

7.2.4 Mecánicas jugables

Las mecánicas jugables propuestas originariamente se han podido implementar. Se pueden piratear drones y estructuras, recupera salud y munición, y la zona jugable está limitada por la mecánica de cobertura de las torres de repetición.

7.2.5 Diseño de niveles

En cuando al diseño de niveles, se ha llegado más lejos de lo inicialmente planeado. Originariamente solo se pensaba realizar un escenario de demo para la segunda entrega y posteriormente un solo escenario jugable. A recomendación de los profesores, se realizó un nivel de tutorial, el cual se ha llegado a implementar junto al resto (el escenario de demo no forma parte del juego final, aunque se puede jugar vía editor).

7.2.6 Control de los personajes e interacción con el juego

Se ha implementado un sistema de control que permite al jugador controlar de manera efectiva los drones y demás aspectos como la cámara.

7.2.7 Flujo del juego

Se ha definido el flujo de pantallas del juego, des de la pantalla de inicio, los niveles del juego y la pantalla de final del juego.

7.2.8 Interfaz de usuario

Se ha diseñado un HUD que permite al jugador controlar tanto los drones como saber las diferentes variables (ver 6.2 HUD). El alumno reconoce que visualmente podría ser más visualmente atractiva, pero el alumno no dispone de los conocimientos de diseño y el tiempo para mejorarlo.

7.2.9 Documentación

Toda la documentación necesaria para las diferentes entregas se ha realizado a tiempo y dentro de los requerimientos establecidos para ello.

7.2.10 Tareas de refinamiento / corrección

A lo largo del desarrollo se han corregido números bugs y se han redefinido diversos aspectos de la jugabilidad. Entre ellas destacan errores de físicas, efectos o IA. En cuanto a las redefiniciones se han modificado diversos aspectos jugables durante el desarrollo tales como valores de cantidad de salud, daño, munición, rutas de drones de enemigos, área de efectos de los ataques, etc.

7.3 Sobre la planificación y metodología seguida

Como se comentó en el punto 1.4 Planificación del Trabajo, se planificaron para el desarrollo múltiples tareas a desarrollar durante las diversas entregas de este trabajo de fin de master.

Como se comentó, la planificación no se llegó a seguir rígidamente, pero el desarrollo seguido a sido muy similar, variando algunas fechas o tiempo dedicado.

Respecto a la metodología, como se comentó anteriormente (1.3 Enfoque y método seguido), se usó las diferentes fechas de entrega como entregas de una supuesto metodología ágil. Esto ayudo a planificar las diferentes tareas y fijarse diferentes objetivos.

7.4 Líneas de trabajo futuro

El juego desarrollado puede servir de base para un producto mucho más completo, no solo a nivel jugable, sino a también a un nivel narrativo y estético.

Como se comentó en la historia del juego, este transcurre en un país sumido en una guerra civil, pero actualmente no hay nada a nivel jugable que represente este aspecto narrativo. Se planeó como objetivo extra implementar una mecánica de confrontación entre los drones, habiendo así facciones enfrentadas entre ellos, pero por falta de tiempo no se pudo implementar. Este aspecto de confrontación puede introducir mecánicas nuevas, como tener que decidir a qué facción ayudar en cada situación, a cambio de, por ejemplo, sortear obstáculos de forma más fácil, abrir nuevos caminos en el escenario o la posibilidad de obtener drones más poderosos.

Esta mecánica también expande las posibilidades narrativas del juego, aspecto no desarrollado durante este proyecto. A través de diferentes herramientas narrativas como transmisiones capturadas a las facciones, archivos de texto o cinemáticas, el jugador podrá conocer las motivaciones y el modo de actuar de cada facción, decidiendo a cuál apoyar basándose en aspectos más morales u ideológicos.

También se abre la posibilidad de incorporar un sistema de recompensas de cada facción, siendo inversamente proporcional a la otra facción. Es decir, las ventajas que obtengamos de una facción tienen su desventaja de cara a la facción rival. Por poner un ejemplo, si una facción nos da más poder de ataque, la otra puede responder haciendo que sus estructuras defensivas tengan más alcance.

Otra posible mejora es la incorporación de un árbol de habilidades, junto con un sistema de experiencia que añada valor a las acciones del jugador durante la partida.

También añadir que durante el desarrollo se usaron assets de terceros, haciendo que el juego no tenga una coherencia estética. Esto puede solventar desarrollando assets propios.

Por último, en el TFM solo se incorporó un tipo de dron y estructura de cada especialidad. Por lo que se pueden añadir muchos más con funciones más concretas.

8. Glosario

- TFM: Trabajo de fin de master.
- IA: inteligencia Artificial
- RTT: Real Time Tactics
- HUD: Head-Up Display (información mostrada en pantalla)
- PEM: Pulso Electro Magnético
- CPU: Central Processing Unit
- Pathfinding: Algoritmo para encontrar el camino óptimo a un destino
- Waypoints: Puntos de ruta. Usados como destinos para el pathfinding
- Assets: Elementos jugables que permiten ser importados entre varios proyectos.
- Collider: Elemento del editor Unity 3D que permite detectar la colisión con otro elemento jugable.
- Shader: Programa para efectuar cambios en la iluminación sobre un elemento jugable.
- Nav mesh: Elemento del editor Unity 3D que crea una estructura de grafos sobre un elemento jugable, que permite ser leídos por los nav mesh agent.
- Nav mesh agent: Lee los grafos definidos por el nav mesh para aplicar un algoritmo de pathfinding para encontrar un camino al destino marcado. Tiene en cuenta varias variables como por ejemplo la inclinación, la esquivas de obstáculos o la altura.

9. Bibliografía

1. Documento: Un juego de carreras. Rafael González Fernández, Pierre Bourdin Kreitz. Documentación Máster Videojuegos de la UOC.
2. Documento: First Person Shooter. Rafael González Fernández, Pierre Bourdin Kreitz. Documentación Máster Videojuegos de la UOC.
3. Documento: Inteligencia artificial para videojuegos. Jordi Duch i Gavaldà, Heliodoro Tejedor Navarro, Samir Kanaan Izquierdo, Gerard Escudero Bakx. Documentación Máster Videojuegos de la UOC.
4. Documento: Generación de efectos en tiempo real de procesamiento. Rafael Pérez Vidal. Documentación Máster Videojuegos de la UOC.
5. Web: https://en.wikipedia.org/wiki/Real-time_tactics
6. Web: <https://blog.kalypsomedia.com/en/commandos-2-hd-remaster-faq/>
7. Web: https://en.wikipedia.org/wiki/Shadow_Tactics:_Blades_of_the_Shogun
8. Web: https://en.wikipedia.org/wiki/Desperados:_Wanted_Dead_or_Alive
9. Web: <https://forum.unity.com/threads/4-6-how-to-detect-if-any-gui-element-has-been-clicked-on.263473/>
10. Web: <https://forum.unity.com/threads/object-jumps-to-another-location-when-enabling-a-navmeshagent.513916/>
11. Web: <http://darrellbircsak.com/2017/01/27/split-unity-terrain-script/>
12. Vídeo: https://www.youtube.com/watch?v=RInUu1_8aGw
13. Vídeo: <https://www.youtube.com/watch?v=cfjLQrMGEb4>
14. Vídeo: <https://www.youtube.com/watch?v=UKs1qO8w7qc&t=226s>
15. Vídeo: <https://www.youtube.com/watch?v=yzcfTb9e0lo>
16. Vídeo: https://www.youtube.com/watch?v=0v_H3oOR0aU&t=177s
17. Vídeo: <https://www.youtube.com/watch?v=iGAdaZ1ICaI>

10. Anexos

10.1 Problemas durante el desarrollo

Durante el desarrollo del juego surgieron dudas sobre cómo implementar ciertos aspectos. Muchos de ellos se solventaron con la documentación que fue proporcionada por los profesores durante el máster ([1], [2]y [3]).

Los problemas más puntuales pudieron ser solventados gracias a los foros dedicados al desarrollo en Unity 3D, como, por ejemplo, las situaciones descritas en los temas [9] y [10]. También se consultaron varios vídeo-tutoriales para solventar dudas puntuales ([12], [13] y [14]), por ejemplo hacer que los misiles persigan objetivos o problemas con la navegación de la IA [16].

Mención especial tiene la implementación de la niebla de guerra, la cual pasó de ser un elemento físico al que se le cambiaba el color en función de sus vértices [17] a ser un shader [15].