

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en ingeniería informática

TRABAJO FIN DE GRADO

**Entrenamiento de drones para la monitorización de incendios
mediante aprendizaje por refuerzo**

Alejandro Cirugeda Pablos
Tutor: Juan Jesús Roldán Gómez
Ponente: Iván Cantador Gutiérrez

Junio 2021

Entrenamiento de drones para la monitorización de incendios mediante aprendizaje por refuerzo

AUTOR: Alejandro Cirugeda Pablos
TUTOR: Juan Jesús Roldán Gómez

Dpto. de Ingeniería informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2021

Resumen

Año tras año hemos observado un aumento considerable del número de incendios producidos por todo el globo. Estos incendios dejan tras de sí numerosas pérdidas tanto materiales como humanas. Debido a la naturaleza estocástica de las llamas, la necesidad de obtener información precisa y en tiempo real es clave para la toma de decisiones en las tareas de extinción de incendios. Pero no es tarea fácil debido a las grandes magnitudes que pueden alcanzar algunos incendios y en ocasiones, la falta de medios personales y materiales. A pesar del uso de vehículos en este tipo de tareas, suponen un número muy reducido con un elevado coste de uso. Para solucionarlo se propone el uso de drones autónomos.

Este Trabajo Fin de Grado tiene como objetivo estudiar la viabilidad de un sistema de monitorización de incendios haciendo uso de drones. Se ha optado por el uso de una arquitectura basada en comportamientos en donde en vez de codificar el sistema global se codifican módulos más sencillos que al ser interconectados logran conductas más complejas al sistema. Además, para la codificación de estos comportamientos se han utilizado técnicas de aprendizaje por refuerzo para la obtención de funcionamientos más elaborados.

Los algoritmos se han desarrollado y validado mediante un entorno de simulación de incendios forestales realista desarrollado en el propio trabajo. Los resultados muestran cómo las aeronaves pueden realizar un seguimiento de la expansión del incendio obteniendo información con un alto grado de fiabilidad respecto al crecimiento del incendio real. Simulaciones adicionales demuestran que el planteamiento se puede escalar aumentando el número de aeronaves y la generalización del conocimiento al poder ser aplicado en diferentes siluetas de incendio.

Palabras clave

Drones, Incendios forestales, Monitorización, Simulación incendios, Arquitectura basada en comportamientos, Aprendizaje por refuerzo, Redes neuronales

Abstract

Year after year we have observed a considerable increase in the number of fires around the globe. These fires leave behind numerous losses, both material and human. Due to the stochastic nature of flames, the need for accurate, real-time information is key to firefighting decision making. But it is not an easy task due to the large magnitudes that some fires can reach and, on occasions, the lack of human and material resources. Despite the use of vehicles in this type of tasks, they represent a very reduced number with a high cost of use. To solve this problem, the use of autonomous drones is proposed.

The objective of this bachelor's Thesis is to study the viability of a fire monitoring system using drones. We have opted for the use of a behaviour-based architecture where instead of coding the overall system, simpler model is coded that when interconnected achieve more complex behaviours. In addition, for the codification of these behaviours, reinforce learning techniques have been used to obtain more elaborate behaviours.

The algorithms have been developed and validated using a realistic wildfire simulation environment developed in this work. The results show how the aircrafts can track the fire expansion obtaining information with a high degree of reliability compared to the growth of the real fire. Additional simulations show that the approach can be scaled up by increasing the number of aircraft and the generalization of knowledge by being able to be applied to different fire shapes.

Keywords

Drones, Wildfires, Monitoring, Wildfire Simulations, Behaviour-based architecture, Reinforcement learning, Neural networks.

Agradecimientos

Agradecer a todas las personas que me han apoyado y estado conmigo en los mejores y peores momentos.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación	1
1.2	Objetivos	3
1.3	Organización de la memoria	4
2	Estado del arte	5
2.1	introducción.....	5
2.2	Robótica	5
2.2.1	Uso de robots en la lucha contra el fuego.....	6
2.2.2	Arquitectura basada en comportamientos.....	7
2.2.3	Aprendizaje por refuerzo	11
3	Diseño y desarrollo.....	13
3.1	Introducción	13
3.2	Recursos	13
3.2.1	Unity	13
3.2.2	ML-Agents.....	14
3.3	Metodología	15
3.4	Campo de entrenamiento.....	16
3.5	Drones	19
3.5.1	Sensores	20
3.5.2	Arquitectura comportamientos	21
3.6	Comportamientos	23
3.6.1	Bordear Fuego	23
3.6.2	Ir batería.....	24
3.6.3	Evitar obstáculos.....	25
3.6.4	Planificador y módulo de fusión.....	26
4	Pruebas y resultados	29
4.1	Introducción	29
4.2	Resultados de cada comportamiento	29
4.3	Resultados sistemas global.....	31
4.1	Otros experimentos	34
4.1.1	Variando el número de drones	34
4.1.2	Formas Irregulares	35
5	Conclusiones y trabajo futuro.....	37
5.1	Conclusiones	37
5.2	Trabajo futuro.....	37
	Referencias	39
	Glosario	41
	Anexos.....	- 1 -
	A Diagrama de clases	- 1 -
	B Material Adicional.....	- 2 -

INDICE DE FIGURAS

<i>FIGURA 1.1: EVOLUCIÓN DE LA SUPERFICIE FORESTAL AFECTADA (2007 - 2017). FIGURA EXTRAÍDA DE [2].</i>	1
<i>FIGURA 1.2: RELEVANCIA DE LOS PROBLEMAS EN TAREAS DE EXTINCIÓN. FIGURA EXTRAÍDA DE [6].</i>	2
<i>FIGURA 1.3: OPINIÓN DE NUEVAS TECNOLOGÍAS EN TEN LA LUCHA CONTRA EL FUEGO. FIGURA EXTRAÍDA DE [6].</i>	3
<i>FIGURA 2.1: DIVERSOS ROBOTS AUTÓNOMOS. A): SPOT, BOSTON DYNAMICS. B): OZ, NAIIO TECHNOLOGIES. C) DRON, DRONES CANARD.</i>	5
<i>FIGURA 2.2: COMPARACIÓN DEL FLUJO DE INFORMACIÓN ENTRE IN MODELO CLÁSICO Y EN BBR FIGURA EXTRAÍDA DE [17].</i>	8
<i>FIGURA 2.3: ROBOT RAVON.</i>	8
<i>FIGURA 2.4: ROBOT NOMAD-CMU.</i>	8
<i>FIGURA 2.5: MODULO BÁSICO IB2C. FIGURA EXTRAÍDA DE [17].</i>	9
<i>FIGURA 3.1: ESQUEMA ESTRUCTURA DE UN ENTORNO EN ML-AGENTS. FIGURA EXTRAÍDA DE [22].</i>	14
<i>FIGURA 3.2: EJEMPLO RUIDO PERLIN.</i>	17
<i>FIGURA 3.3: EJEMPLO EVOLUCIÓN INCENDIO.</i>	18
<i>FIGURA 3.4: MODELO 3D DE UN DRON.</i>	19
<i>FIGURA 3.5: SENSOR UST-20LN.</i>	20
<i>FIGURA 3.6: ESQUEMA DE LAS CELDAS DETECTADAS POR EL DRON.</i>	21
<i>FIGURA 3.7: ESQUEMA ARQUITECTURA DEL SISTEMA.</i>	22
<i>FIGURA 3.8: ESTRUCTURA DETERMINISTA DEL COMPORTAMIENTO.</i>	25
<i>FIGURA 3.9: ESQUEMA SIMPLIFICADO DEL MODULO DE PLANIFICACIÓN.</i>	26
<i>FIGURA 3.10: ENTORNO DE UNITY ENTRENANDO EL COMPORTAMIENTO DE BORDEAR FUEGO.</i>	27
<i>FIGURA 4.1: EVOLUCIÓN DE RECOMPENSAS A LO LARGO DEL TIEMPO.</i>	30
<i>FIGURA 4.2: EVOLUCIÓN LONGITUD EPISODIO A LO LARGO DEL TIEMPO.</i>	31
<i>FIGURA 4.3: TRAYECTORIAS DE LOS DRONES EN LA SIMULACIÓN.</i>	32
<i>FIGURA 4.4: PORCENTAJE DE SIMILITUD DEL INCENDIO REAL POR EL DETECTADO POR LOS DRONES.</i>	33
<i>FIGURA 4.5: GRAFICA DE LA SIMULACIÓN CON 3 Y 4 DRONES.</i>	34

FIGURA 4.6: GRAFICA DE LA EVOLUCIÓN DEL INCENDIO CON FORMA DE U. 35

FIGURA 4.7: GRAFICA DE LA EVOLUCIÓN DE UN INCENDIO EN FORMA DE LÍNEA. 35

FIGURA 4.8: EVOLUCIÓN INCENDIO EN FORMA DE T...... 36

INDICE DE TABLAS

TABLA 1: PROYECTOS CONTRA INCENDIOS CON USO DE AERONAVES 7

1 Introducción

1.1 Motivación

El impacto de los incendios forestales aumenta cada año, y se han convertido en un desafío tanto ambiental como económico que deja cada año cientos de víctimas mortales, miles de evacuados e innumerables daños materiales y ecológicos. En este último lustro hemos visto incendios devastadores desde la selva Amazónica, la costa californiana, o los terribles incendios de Australia. Debido al cambio climático y la despoblación de zonas rurales cada año se incrementa el número de grandes incendios forestales, también denominados GIF (aquellos que afectan a una superficie superior a 500 hectáreas), cada vez más virulentos e imposibles de controlar por medios de extinción que provocan crisis nacionales. Estos grandes incendios forestales apenas suponen el 0,18% del total de incidencias registradas, pero en ellos se ve afectada un 40% de la superficie total afectada [1].

En 2019, En España hubo un total de 10.883 incendios con una superficie quemada de 83.963ha. Con una media anual de 98.000ha entre 2010-2020, la cual supone un 0,342% de la superficie española cada año [3]. Si contamos los grandes incendios forestales, en España se produce una media anual de 12 grandes incendios forestales con una superficie aproximada de 60.000 hectáreas. [4]. Donde destaca el incendio de la isla de Gran Canaria que devoro 10.000 hectáreas en 2019 y afecto al 6,5% de la superficie de la isla. Estos datos sitúan a España como el segundo país del mediterráneo donde más superficie arde cada año [1].



Figura 1.1: Evolución de la superficie forestal afectada (2007 - 2017). Figura extraída de [2].

Estos incendios no suponen únicamente daños al patrimonio ambiental o rural, sino que afecta un grave riesgo para la vida de las personas, tanto de residentes de las zonas afectadas como el personal que trabaja en las tareas de extinción. La tarea de la extinción de incendios es una operación peligrosa debido a la aleatoriedad del incendio y que puede provocar la muerte a los profesionales que intervienen en ella. El gobierno español ha reportado 37 fallecimientos de bomberos entre 2006 y 2015 [5] y solo en 2017 los grandes

incendios de España y Portugal se registraron en la región incendios de alta intensidad que dejaron cientos de víctimas mortales.

En la publicación [6] se llevaron a cabo una serie de entrevistas a un grupo de bomberos forestales con el objetivo de obtener información del estado actual de la lucha contra el fuego, de los principales problemas en las diferentes tareas y sobre las opiniones del uso de nuevas tecnologías para solucionar estos problemas.

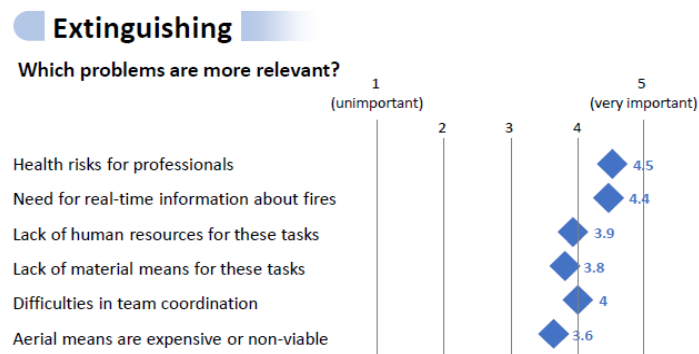


Figura 1.2: Relevancia de los problemas en tareas de extinción. Figura extraída de [6].

En la Figura 1.2 podemos observar los problemas más relevantes en la tarea de extinción. Cabe recalcar que la extinción del fuego no solo depende de las propias acciones de apagar las llamas sino también todas aquellas actividades que apoyan esta acción como la creación de cortafuegos, creación de rutas de entrada y salida de personal, obtención de información en tiempo real, etc.

Como podemos ver en los resultados en la encuesta respecto a las tareas de extinción podemos ver como la preocupación más importante es el riesgo para la salud de los profesionales con una puntuación de 4,5 sobre 5 seguido de la necesidad de información en tiempo real de la situación. Estos problemas se encuentran relacionados debido a que un gran número de los fallecimientos producidos se deben al atrapamiento y a caídas [5]. Además, estas encuestas revelan una falta importante de personal y de material y de la importancia de la información en tiempo real del entorno en tareas de extinción. De manera rutinaria se utilizan múltiples tipos de vehículos y maquinaria para mejorar la efectividad y la seguridad de las diferentes operaciones, aunque estos vehículos son escasos y presentan un enorme gasto para cada cuerpo de bomberos y son generalmente controlados por uno o varios profesionales.

En [6] se llevaron también una serie de encuestas para ver la opinión de los profesionales sobre diferentes tecnologías para poder solucionar los diferentes problemas comentados anteriormente.

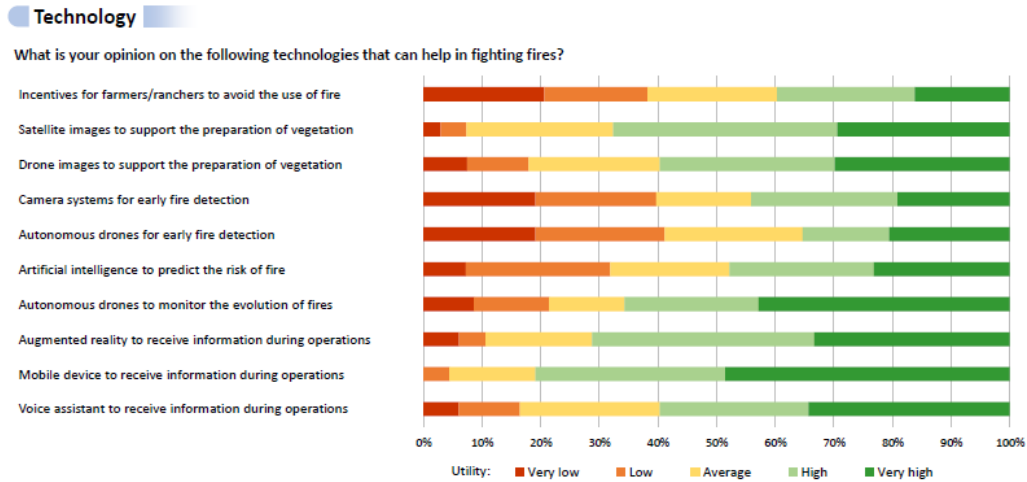


Figura 1.3: Opinión de nuevas tecnologías en ten la lucha contra el fuego. Figura extraída de [6].

En la Figura 1.3 podemos ver los resultados de las opiniones de sobre el uso de la tecnología. Estos resultados muestran que dos de las tecnologías con mejor opinión son: en primera posición, el uso de dispositivos para la notificación de la evolución del fuego y, en tercera posición, el uso de drones para la monitorización de las llamas. Estas respuestas tienen mucho sentido al ser la propagación de las llamas a un evento muy irregular y estocástico que puede variar en cada momento.

Debido a todo lo comentado anteriormente, tiene sentido la creación de una solución tecnológica que pudiese solucionar los problemas tanto de escasez de personal como de la falta de información en tiempo real en las operaciones de extinción. Pudiendo reducir tanto los índices de accidentalidad como la mortalidad en este tipo de tareas y de facilitar las tareas de extinción de las llamas para los profesionales.

1.2 Objetivos

El objetivo de este proyecto es estudiar la viabilidad de un sistema de monitorización de un incendio forestal basado en un enjambre de drones con el objetivo de obtener un seguimiento en tiempo real de la expansión de las llamas. Para ello se utilizará una arquitectura basada en comportamientos junto a la utilización de técnicas de aprendizaje por refuerzo para su desarrollo. Se han dividido en diferentes objetivos:

- El desarrollo de un entorno de simulación donde se puedan producir incendios y que se produzca su expansión a lo largo del tiempo.
- El Diseño de arquitectura para el dron basado en comportamientos.
- El entrenamiento de la arquitectura para su despliegue en el campo de simulación.
- La obtención de una arquitectura funcional que obtenga un alto porcentaje de similitud entre la información obtenida por los drones respecto al incendio original.

Para la comprobación de estos objetivos se ha establecido una serie de requisitos del proyecto que serán analizados posteriormente para ver el grado del cumplimiento de estos:

- Obtención de un grado alto de fiabilidad del modelo de expansión del fuego que permita la múltiple modificación de parámetros como la velocidad de expansión o la intensidad y dirección del viento.
- Comprobación del grado de funcionalidad de manera independiente de los comportamientos.
- Realización y comprobación de un sistema de interacción entre los diferentes comportamientos.
- En la simulación de un incendio, la obtención de un grado de fiabilidad alto (80%) de manera continuada entre la información aportada por los drones y la situación real del incendio.
- Demostración del aprendizaje de los agentes comprobando si han podido generalizar el problema a resolver.

1.3 Organización de la memoria

La memoria se estructura de la siguiente manera:

- Capítulo 2: Estado del arte. Se explica la situación actual y el estado del arte de los fundamentos en los cuales se basa el proyecto, hablando sobre el uso de robots autónomos en diferentes sectores en la actualidad y haciendo hincapié en los proyectos de investigación de su uso contra incendios.
- Capítulo 3: Diseño y Desarrollo. Descripción de los diferentes recursos que se han utilizado para la realización de este proyecto seguido del diseño y estructura del sistema desarrollado, junto con las decisiones de diseño tomadas. Además del desarrollo del sistema profundizando en cada uno de los módulos desarrollados.
- Capítulo 4: Pruebas y resultados. Exposición de las diferentes simulaciones y desempeño del sistema.
- Capítulo 5: Conclusiones y trabajo futuro. Cierre del proyecto donde se analiza el cumplimiento de los objetivos propuestos y de las líneas de investigación que se han abierto.

2 Estado del arte

2.1 introducción

Debido a la variedad de ramas y disciplinas exploradas durante este proyecto, es necesario realizar un estudio del estado del arte con el objetivo de situar el proyecto dentro de cada rama y mencionar las tecnologías y métodos que se utilizan en la actualidad. Así pues, en este apartado se comentará:

- El estado actual del campo de la robótica.
- Uso de robots en tareas de monitorización y extinción de incendios.
- Arquitecturas basadas en comportamientos.
- Aprendizaje por refuerzo.

2.2 Robótica

En esta última década se ha podido observar un enorme avance en el campo de la robótica, haciendo que cada día veamos robots más complejos, precisos y robustos. Por otra parte, gracias al avance de la tecnología de nuevo tipos de sensores. La tecnología LIDAR, los avances en las técnicas de procesamiento de imágenes y los sensores pueden aportar a la máquina numerosos estímulos sobre su entorno como obstáculos, temperatura, luz y así facilitar al robot a obtener un mapeo verídico de sus alrededores. Especialmente la última revolución en el campo de la robótica ha tenido que ver con los grandes avances dentro del sector de la Inteligencia artificial (IA), los cuales les han permitido obtener un grado de autonomía e interacción con el entorno que antes era inimaginable.

Esta implementación de la inteligencia artificial junto a la robótica ha aportado unos resultados admirables, desde coches autónomos, aeronaves no tripuladas o incluso control de brazos mecánicos. Por esta razón estamos actualmente viviendo una revolución donde cada año se empiezan a combinar más estas dos disciplinas. Hoy en día el camino que persigue la robótica es conseguir los sistemas más autónomos posibles donde no se requiera interacción humana. Estos avances han facilitado la implementación de sistemas inteligentes en una gran variedad de sectores.



Figura 2.1: *Diversos Robots autónomos. A): Spot, Boston Dynamics. B): OZ, NAIO Technologies. C) Drone, Drones Canard.*

En la Figura 2.1 podemos observar diversos robots autónomos. Spot, robot de la empresa Boston Dynamics es un robot comercial disponible desde 2019 que es utilizado para

diversas tareas desde la inspección en procesos industriales peligrosos, transporte de carga e incluso desactivación de bombas en escuadrones de policía. En otros campos como la agricultura también existen actualmente robots autónomos comerciales como el caso de la empresa NAIIO que tiene en el mercado 3 modelos de robots autónomos encargados de tareas como detección y eliminación de malas hierbas, siembra de semillas y dando apoyo en la creación de surcos en los terrenos. Asimismo, existen aeronaves autónomas comerciales como en el caso de la compañía Canard los cuales disponen de un servicio autónomo de inspección de pistas de aterrizaje. Queda claro que en los próximos años aparecerán numerosas empresas que ofrecerán nuevas soluciones tecnológicas en diversos sectores con robots autónomos, como puede ser en la lucha contra el fuego.

2.2.1 Uso de robots en la lucha contra el fuego

En la lucha contra incendios, la mayoría de los robots que se utilizan en la actualidad son controlados de manera remota por bomberos que se encuentran en una zona segura alejada de las llamas. Estos robots pueden ser tanto aéreos como terrestres y realizan numerosas tareas como el transporte de materiales, apagado de incendios y la obtención de imágenes para los recursos humanos desplegados en la zona afectada. Debido a los grandes avances en el campo de la inteligencia artificial, los vehículos para la lucha contra incendios del futuro serán autónomos. La intención del uso de vehículos no tripulados con el objetivo de ayudar en tareas de extinción de incendios no es una idea novedosa y actualmente existen numerosos proyectos de investigación vigentes con gran variedad de uso de vehículos como UAV, planeadores, helicópteros o drones para diferentes tareas.

En la Tabla 1 se enumeran los trabajos más importantes. Podemos observar a la gran variedad de proyectos de investigación que han experimentado con aeronaves reales y en situaciones controladas. Como se puede observar, destacan por su gran variedad de objetivos, de vehículos que utilizan y diferentes técnicas que utilizan. Existen numerosos tipos de vehículos que se pueden utilizar para cada tipo de tarea, cada uno con sus ventajas e inconvenientes. El uso de drones para la monitorización en tiempo real de un incendio forestal es ideal debido a su bajo coste, su fácil transporte y fácil escalabilidad en la implantación de sistemas.

Proyecto	Objetivo	Vehículo	Autónomo	IA
WildHopper [7]	Creación de cortafuegos transportando hasta 600 litros de agua	Drones pesados (600kg)	Si	Sin especificar
Sky-eye [8]	Detección y monitorización	UAV ala fija y satélites	No, control remoto	No
University of seville [9]	Monitorización y obtención de datos de un incendio	UAV ala fija y pequeños helicópteros	Si, a través de movimiento predefinido y responde a peticiones	No
Altus [10]	Obtención datos tiempo real	UAV ala fija	Si, con waypoints predefinidos	No
RFSAT -unarmed department [11]	Identificación zonas potenciales de incendios y modelado de la zona	Drones	Si, con waypoints predefinidos	No
Sevilla, Urban Fires [12]	Detección de incendios en entornos urbanos	UAV ala y cámaras fijas	Si	Si, detección visual de humos
Fire [13]	Medición de datos del incendio como variación de viento y temperatura	Drones	No, control remoto	Modelo de evaluación de humo
Ikhana [14]	Obtención de imágenes de infrarrojos	UAV y satélites	No control remoto	No
VALAQ Patrol [15]	Patrulla y detección de incendios	UAV ala fija	No, control remoto	No
Proyecto Telefónica - Carlos III [16]	Detección temprana y prevención de incendios	Drones y sensores fijos	Si	Sin especificar

Tabla 1: Proyectos contra incendios con uso de aeronaves.

En el proyecto actual a desarrollar se pretende conseguir un sistema totalmente autónomo que utilice un conjunto de drones para conseguir información en tiempo real del estado del incendio. Esto se lleva a cabo utilizando una arquitectura basada en comportamientos y técnicas de aprendizaje automático para la codificación de los comportamientos más complejos. A pesar de ser un proyecto que se desarrolla en un entorno de simulación, la intención es de implementarlo en drones reales.

2.2.2 Arquitectura basada en comportamientos

El control basado en comportamientos o BBR (Behaviour Based Robotics) es una metodología de diseño que se utiliza en numerosos sistemas de robots complejos. Esta arquitectura se basa en la separación de diferentes módulos o comportamientos que se encuentran interconectados cooperando, pero manteniendo una independencia propia. Esto presenta una serie de ventajas como la no dependencia de un único modulo y la robustez que presenta ante información desconocida gracias a la interacción entre los distintos comportamientos. Con este tipo de arquitecturas se pueden obtener una actuación global compleja a pesar de simplicidad de la codificación de sus comportamientos internos [17].

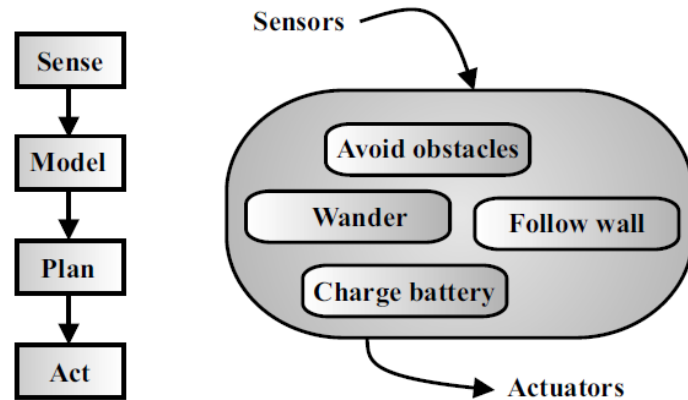


Figura 2.2: Comparación del flujo de información entre un modelo clásico y en BBR
Figura extraída de [17].

En la Figura 2.2 podemos ver la comparativa entre la estructura del modelo clásico de aprendizaje automático y un sistema BBR basado en comportamientos. La estructura clásica de la IA está basada en un razonamiento complejo de alto nivel como puede ser el reconocimiento de patrones. Mientras que la estructura basada en comportamientos es una alternativa al sistema clásico en donde se construye a partir de comportamientos sencillos los cuales pueden estar ejecutándose de manera paralela dando diferentes sugerencias de las acciones que el robot tiene que tomar.

Las arquitecturas de BBR suelen ser implementadas en robots autónomos, los cuales no requieren de supervisión humana y se pueden manejar de manera libre. En la actualidad existen numerosos Robots con este tipo de arquitecturas como el Robot RAVON (Robust Autonomous Vehicle for Off-road Navigation) (Figura 2.3) desarrollado en el Laboratorio de Robótica de la Universidad de Kaiserslautern, Alemania [18]. El objetivo de este robot es tener un vehículo autónomo que pueda atravesar cualquier tipo de terreno en tareas en escenarios hostiles como situaciones de accidentes o desastres naturales.



Figura 2.3: Robot Ravon.



Figura 2.4: Robot Nomad-CMU.

En la figura 2.4 podemos observar otro ejemplo de vehículo basado en comportamientos para la exportación autónoma de terrenos áridos desarrollado en la Instituto de robótica la universidad Carnegie Mellon en Pensilvania [19].

Para este proyecto se usa el diseño IB2C [17], una arquitectura basada en comportamientos publicada en 2010 en la universidad Kaiserslautern, Alemania para el diseño de robots complejos. Como se ha comentado, este tipo de arquitecturas se encuentran formadas por módulos de comportamientos. Estos comportamientos mediante la información que reciben como entrada, decretan el estado en el que se encuentra y deciden la salida del módulo. Debido a que cada comportamiento indica una posible acción, la arquitectura tiene diferentes mecanismos de coordinación entre los comportamientos para que puedan influenciarse entre ellos. Pasamos a enumerar las entradas y salidas del módulo básico de comportamiento.

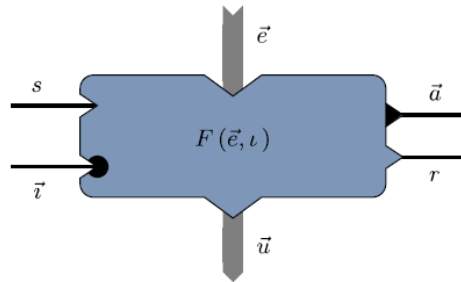


Figura 2.5: Módulo básico IB2C. Figura extraída de [17].

Las entradas del comportamiento son:

- **Vector de entrada e :** Es el vector de entrada con la información necesaria para que el comportamiento pueda generar las acciones adecuadas. Esta información puede venir tanto del entorno mediante sensores como de otros comportamientos de la arquitectura.
- **Estimulación s :** La señal s representa el grado de relevancia del comportamiento en ese instante de tiempo. Es un valor comprendido entre $[0,1]$ siendo $s = 0$ la estimulación nula y $s = 1$ la máxima estimulación.
- **Vector de inhibición i :** La inhibición i reduce la relevancia de la señal de estímulo s . De esta manera un comportamiento puede estar inhibido por diferentes módulos. El valor se encuentra comprendido entre $[0, 1]$ siendo 0 la nula inhibición y 1 la inhibición completa del estímulo del comportamiento.
- **Activación u :** Esta señal determina la relevancia efectiva del comportamiento dentro de la arquitectura teniendo en cuenta el vector inhibición. La señal a se calcula como:

$$u = s * (1 - i) \quad \text{Ecuación 1}$$

Y las salidas del comportamiento son:

- **Vector Salida u :** Vector de salida es la acción tomada por el comportamiento, decidida por el vector entrada e . Este vector de acción puede ser transmitida a actuadores o a otros comportamientos.
- **Actividad a :** La actividad indica el grado de influencia del comportamiento dentro del sistema en el instante de tiempo. Es un valor comprendido entre $[0,1]$ siendo $a =$

0 indica el nulo impacto del vector u en el sistema total y $a = 1$ indica un gran impacto del vector u .

- **Calificación objetivo r :** Indica el grado de satisfacción del comportamiento según el estado en el que se encuentre el sistema. El valor está comprendido entre $[0,1]$ siendo $r = 0$ indica que el comportamiento está completamente satisfecho y $r=1$ se encuentra completamente insatisfecho.

Módulos de fusión

En el desarrollo de arquitecturas basadas en comportamientos es necesario una coordinación entre los diferentes módulos que forman el sistema. Esto supone uno de los principales problemas de este tipo de sistemas. La coordinación entre los comportamientos se logra mediante la unificación de sus salidas, en un nuevo módulo denominado módulo de fusión. La estructura de este módulo es la misma que presenta el módulo básico de comportamiento explicado anteriormente.

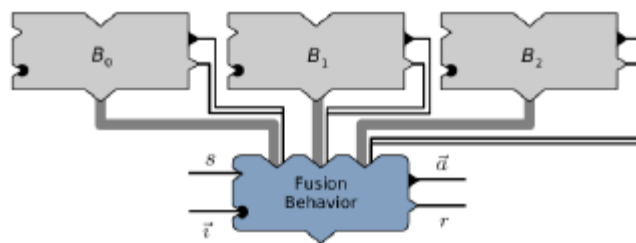


Figura 2.8: Ejemplo de módulo de fusión. Figura extraída de [20].

El objetivo de este módulo es fusionar los valores del vector salida u de cada comportamiento, teniendo en cuenta la señal *activity* de cada comportamiento. Existen diversas maneras de determinar la salida final del módulo, desde obtener la entrada con el valor de *activity* más alta, hasta la suma ponderada de las entradas.

Las arquitecturas basadas en comportamientos requieren un mecanismo que coordine la ejecución de los diferentes comportamientos. El desarrollo de estas técnicas dentro de la arquitectura IB2C se consigue entre la combinación adecuada ente el módulo de fusión y la interacción entre los comportamientos. Además, este tipo de arquitecturas permite el uso de las diferentes técnicas en diferentes capas, teniendo así un sistema que implemente diferentes técnicas. Se podría destacar 3 tipos de técnicas de coordinación:

- **Priority-based:** Se implementa mediante la inhibición de los comportamientos de manera sucesiva, de modo que el orden de los comportamientos determina el grado de prioridad de cada uno de ellos. El módulo de fusión seleccionará aquel comportamiento que presente una mayor señal de activación.
- **Stated-based-mechanism:** La coordinación de los comportamientos se consigue mediante el módulo de fusión de máxima activación. Este mecanismo cuenta con un módulo de evaluación de estados, encargado de evaluar el estado actual del entorno y estimular a los diferentes comportamientos.
- **Winner-takes-all mechanism:** Los comportamientos se fusionan únicamente utilizando el módulo de fusión.

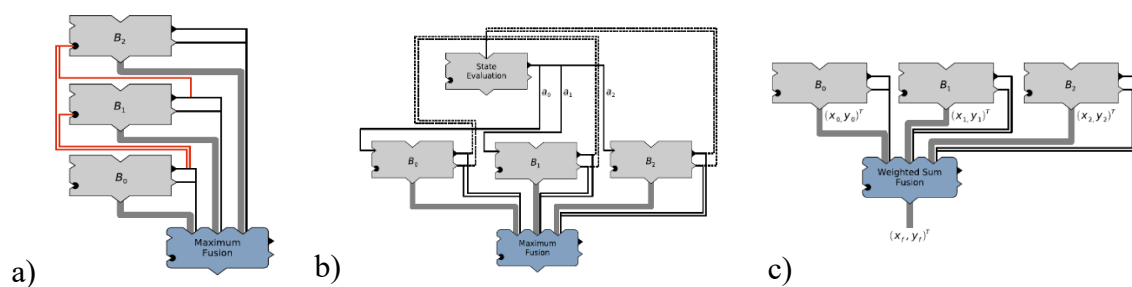


Figura 2.9: Diferentes técnicas de coordinación. A) Arquitectura “priority-based”. B) Arquitectura “based-mechanism”. C) Arquitectura “winner-takes-all-mechanism”.
Figura extraída de [20].

En ocasiones, estos comportamientos pueden ser complicados de codificar debido a su gran casuística que pueden presentar y es por eso que se pueden utilizar técnicas de aprendizaje automático como aprendizaje por refuerzo con el objetivo de poder adaptarse a nueva entrada de datos.

2.2.3 Aprendizaje por refuerzo

El aprendizaje por refuerzo es un subcampo del aprendizaje automático (Machine Learning) donde el sistema a aprender, también llamado agente, aprende mediante la interacción con su propio entorno, observando o el resultado de sus acciones y recibiendo una recompensa o castigo de acorde a su acción. Este tipo de aprendizaje es similar al aprendizaje que desarrollan tanto los humanos como de animales.

El aprendizaje por refuerzo involucra aprender qué hacer con el objetivo de maximizar el valor de señal de recompensa. Estos sistemas de aprendizaje son de lazo cerrado debido a que las acciones del agente afectan a las futuras entrada de datos. Y a diferencia de otras técnicas de aprendizaje automático, como el aprendizaje supervisado, al agente no se le indica las acciones que debe tomar, sino que debe descubrir el conjunto de acciones que le lleven a maximizar la recompensa con prueba y error [21].

La gran diferencia entre el aprendizaje supervisado y el aprendizaje por refuerzo es que el aprendizaje supervisado aprende de un conjunto de datos etiquetados aportados por una supervisión externa, la cual indica la acción que tenía que haber tomado el sistema en esa situación, generalmente indicando a que categoría pertenece. Este tipo de aprendizaje es utilizado con la intención de extrapolar y generalizar la información para que el sistema responda de manera correcta en situaciones que no pertenezcan al conjunto de datos inicial. Además, el aprendizaje por refuerzo también se diferencia del aprendizaje no supervisado, los cuales se basan típicamente en encontrar patrones desconocidos dentro de un conjunto de datos sin etiquetar. La diferencia entre estas dos técnicas de aprendizaje automático es que el aprendizaje por refuerzo tiene como objetivo maximizar la función de recompensa en vez de encontrar patrones desconocidos.

Dadas estas diferencias, el aprendizaje por refuerzo se puede considerar la tercera técnica de aprendizaje automático diferente al aprendizaje supervisado y el aprendizaje no supervisado.

Debido a que las acciones del agente repercuten posteriormente en la entrada de información para futuras acciones a tomar, este tipo de aprendizaje se enfrenta a un enorme reto que se encuentra solo en este tipo de aprendizaje, la decisión entre explorar y explotar. Para que el agente pueda encontrar las acciones que maximizan su recompensa, es necesario que utilice estrategias que se hayan demostrado que aporta grandes recompensas en el pasado, tiene que *explotarlas*. Pero además de utilizar y profundizar en las estrategias que aporten recompensas el agente debe también *explorar* el conjunto de acciones para poder tomar mejores decisiones en el futuro. Por lo tanto, para que el agente pueda completar su tarea con éxito es necesario un balance entre *exploración* y *explotación*.

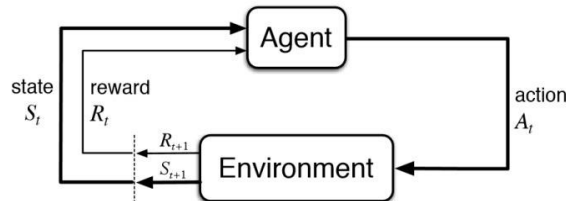


Figura 2.10: Interacción Agente-Entorno en aprendizaje por refuerzo. Figura extraída de [21].

En la Figura 2.10 se puede observar un esquema del aprendizaje por refuerzo. En él, el agente y su entorno interactúan con el entorno cada t tiempo, en el cual el agente recibe información de su entorno S_t . Dependiendo del estado del entorno en el tiempo t el agente decide una acción A_t . Y en el siguiente periodo de tiempo dependiendo la acción tomada recibe una recompensa o castigo R_t .

El aprendizaje por refuerzo en problemas secuenciales se puede definir mediante un proceso de decisión de Markov (MDP). Este proceso se representa mediante una tupla de 5 elementos (S, A, P, R, γ) en donde:

- S es el conjunto de estados.
- A es el conjunto de acciones.
- $P(s, a, s')$ es la probabilidad de la acción a en el estado s en el tiempo t lleve al estado s' en el $t+1$.
- $R(s, a, s')$ es la recompensa inmediata recibida después de la transición del estado s al estado s' , debido a la acción a .
- γ es el factor de descuento de cada recompensa a lo largo del tiempo.

El objetivo del aprendizaje por refuerzo es encontrar la política, el modelo o el conjunto de acciones que maximiza el valor de las recompensas. Debido a que el agente puede aprender sin supervisión experta este tipo de aprendizaje se puede utilizar en tareas donde la codificación y programación de la solución no es sencilla. En problemas donde no se pueda etiquetar de manera clara y concisa la información que recibe como entrada. Dado todo lo anterior es por el cual las técnicas de aprendizaje por refuerzo son fuertemente utilizadas en el campo de la robótica. “El aprendizaje por refuerzo ofrece a la robótica un marco de trabajo y un conjunto de herramientas para el diseño comportamientos sofisticados y difíciles de diseñar.” (J. Kober, J. Andrew Bagnell, & J. Peters) [21]

3 Diseño y desarrollo

3.1 Introducción

La etapa de investigación de los capítulos anteriores ha servido como base para la toma de decisiones del sistema a desarrollar. Se decidió realizar un entorno de simulación en Unity para reproducir la expansión de un incendio y en donde desarrollar y testear diferentes comportamientos para los drones. Se dará una explicación detallada de cada una de las etapas donde se exponen los fundamentos y el procedimiento utilizado para el desarrollo. Los apartados son los siguientes:

- Recursos.
- Campo de entrenamiento.
- Simulación de drones.
- Comportamientos individuales.
- Sistema de coordinación de comportamientos.

3.2 Recursos

En esta sección se hablará de los recursos y herramientas principales utilizadas para el desarrollo del sistema de monitorización.

3.2.1 Unity

Unity es una plataforma que nació en 2016 como un motor 3D para el desarrollo de videojuegos, pero debido a sus capacidades, a sus numerosas librerías y recursos que ofrece es actualmente utilizado en diferentes campos como arquitectura, cinematografías y simulaciones entre muchas otras.

La estructura de los proyectos de Unity consiste en la creación de objetos jerárquicos denominados *GameObjects*. Estos objetos pueden tener asociados diferentes componentes que les aportan diferentes características tales como forma, color, movimiento, gravedad. Con este tipo de cualidades podemos crear infinidad de objetos como aeronaves, terrenos, partículas etc. Unity además de tener componentes ya desarrollados permite la codificación de nuevos mediante scripts generados en C#. Mediante estos scripts se permite lograr comportamientos complejos e interacción entre los diferentes objetos con los cuales estemos trabajando. Asimismo, Unity presenta un amplio catálogo de *Assets* (modelos 3D, imágenes, audios que se encuentran disponibles para su uso libre) y de librerías y herramientas que permiten reducir enormemente el tiempo de desarrollo. El campo del aprendizaje automático, especialmente el aprendizaje por refuerzo ha cogido gran importancia en esta herramienta debido a las numerosas librerías que se ofrecen. Unity es la herramienta ideal para poder generar entornos simulados en donde codificar y entrenar nuestros robots sin los problemas asociados al mundo real como su deterioro con el ambiente.

Este deterioro del hardware viene asociado con un alto coste tanto económico como temporal, debido al tiempo y esfuerzo necesario para su cuidadoso mantenimiento. Además del deterioro, la necesidad de la repetición constante de las experiencias supone otro problema a solucionar. Asimismo, si el campo del entrenamiento necesita algún preparativo

inicial, como la recolocación de los diferentes objetos en su posición inicial, supone un significativo aumento del tiempo del desarrollo. En un entorno virtual permite tener una despreocupación sobre la integridad de los robots que se están desarrollando y un reinicio del entorno de simulación automático disminuyendo considerablemente el tiempo de desarrollo y entrenamiento.

Por último, otro problema que supone el aprendizaje por refuerzo en el mundo real es la capacidad de comparación de diferentes algoritmos, el uso de diferentes configuraciones o incluso durante el propio aprendizaje las dinámicas del robot pueden cambiar debido a numerosos factores externos como el viento, la temperatura o la iluminación.

Debido a estos problemas es necesario crear una simulación del entorno, un campo de entrenamiento donde el agente este exento de los problemas del mundo real y donde pueda interactuar con su entorno sin riesgo alguno. Además, un campo de entrenamiento nos supone una serie de ventajas como poder aumentar la velocidad del sistema reduciendo así el tiempo de entrenamientos.

3.2.2 ML-Agents

Unity Machine Learning Agents [22] (ML-Agents) es una librería *open-source* para Unity, desarrollada por el equipo de investigación de OpenAI, que permite que juegos y simulaciones sirvan como campo de entrenamiento para la investigación y el desarrollo de modelos de inteligencia artificial. Esta plataforma es tanto beneficiosa tanto para desarrolladores de videojuegos como a investigadores de inteligencia artificial, donde los avances en el campo del aprendizaje automático pueden ser evaluados en un entorno rico y accesible para un público más amplio.

La creación de esta librería supuso una aproximación de la inteligencia artificial a numerosos campos y el cual ha supuesto la aparición de numerosas publicaciones en varios sectores como en la robótica, visión computacional y en simulaciones computacionales.

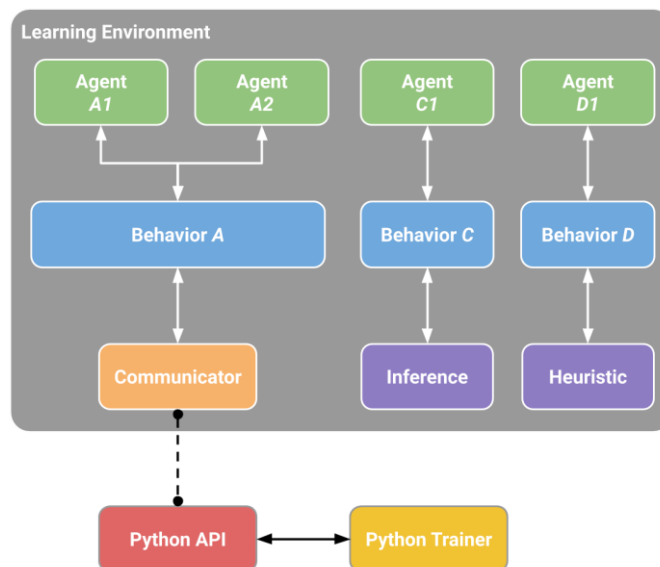


Figura 3.1: Esquema estructura de un entorno en ML-Agents. Figura extraída de [22].

Esta herramienta permite la creación de campos de entrenamiento para nuestros agentes, uniendo los componentes 3D de Unity junto a los modelos de inteligencia artificial. La librería es una implementación basada en Tensorflow y que cuenta con numerosos algoritmos del estado del arte como PPO (*Proximal Policy Optimization*), GAIL (*Generative Adversarial Imitation Learning*) o SAC (*Soft Actor Critics*) y añadiendo nuevos modelos cada versión. En la Figura 3.1 podemos observar la estructura simplificada de un entorno de entrenamiento creado en MI-Agents, en donde hay 4 agentes distintos, representados en la parte superior. Cada agente tiene asociado uno modelo distinto de entrenamiento e incluso pudiendo ser compartido entre varios agentes. El modelo A está siendo entrenado con un algoritmo de aprendizaje por refuerzo realizado en Tensorflow y el cual se comunica mediante un comunicador al entorno de Unity.

Los modelos creados en esta herramienta son redes neuronales. Estas redes están formadas por un conjunto de elementos denominados neuronas que se encuentran interconectados para enviarse señales. Estas neuronas reciben unos valores de entradas, aplican una operación y envían su salida a la entrada de la siguiente neurona. Estas operaciones que realizan las neuronas se van ajustando de manera automática con el objetivo de conseguir la salida deseada.

Esta librería permite la total modificación de estas redes neuronales como la modificación del número de capas, el número de neuronas por capas y los hiperparámetros de cada algoritmo para poder adaptarse a diferentes problemas

Con este *framework* se simplifica en gran medida el desarrollo y entrenamiento de modelos, permitiendo incluso entrenar de manera simultáneos diferentes modelos con diferentes entornos de aprendizaje. Otra gran ventaja que representa es la fácil y rápida modificación de los hiperparámetros de los diferentes modelos a utilizar. Además, aumenta la velocidad de entrenamiento de los modelos al permitir ejecutar de manera simultánea varios campos de entrenamiento aportando información al modelo que se encuentre entrenando.

3.3 Metodología

El trabajo realizado se encuentra estructurado en la creación del campo de entrenamiento, la creación diferentes comportamientos y la unificación de estos para conformar el sistema final. En el anexo I se adjunta el diagrama de clases del sistema.

Para la realización de este trabajo se ha empleado una metodología incremental, donde inicialmente se plasmó la idea con pequeños esquemas y bocetos y posteriormente se obtuvieron los requisitos funcionales y no funcionales tanto del sistema global como de los diferentes módulos. Con el desarrollo incremental se fueron creando los diferentes incrementos y añadiéndolos al sistema hasta tener el proyecto final.

3.4 Campo de entrenamiento

Debido a las limitaciones comentadas anteriormente en la sección 2 con el aprendizaje automático en la robótica se ha decidido realizar un campo de entrenamiento en el cual se simule un incendio intentado replicar el mayor realismo posible. Este entorno de simulación permite la libre experimentación asegurando la integridad de las aeronaves y ahorrando numerosos costes de hardware.

Actualmente en el mercado existen diferentes simuladores de incendios y transferencias de calor utilizados para tareas de investigación o de seguridad industrial. Estas herramientas utilizan diferentes técnicas de modelado en 2 y 3 dimensiones como simulación de fluidos, sistemas deterministas o sistemas físicos [23]. Estas herramientas suelen reflejar datos muy similares al mundo real que vienen asociados con un gran uso de poder computacional y gran coste de licencias software. En este trabajo se ha optado por la realización de una simulación de incendio con Unity desde cero basadas en ecuaciones diferenciales para la transmisión de calor [24].

Para ello se ha generado un terreno en Unity en donde se le ha podido aplicar un *script* donde se modeliza el incendio. Esto nos permitirá tener un control absoluto pudiendo ajustar numerosos parámetros como tamaño del incendio, velocidad de propagación, velocidad del viento... El terreno generado se ha discretizado y generado una cuadrícula de celdas de dimensión fija. De esta manera, cada celda del terreno tendrá su propia temperatura y combustible y aplicando ecuaciones de transmisión de calor en 2D se modelizará la difusión y expansión de temperatura entre las celdas provocando así un incendio. En la ecuación 2 podemos observar la ecuación del calor aplicado a 2 dimensiones en donde se muestra que todo el calor producido en un punto se dispersa en el espacio.

$$\frac{\partial T}{\partial t} - \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0 \quad \text{Ecuación 2}$$

En la ecuación 2 se indica como la temperatura T a lo largo del tiempo se dispersa por el espacio en las coordenadas x e y controlado por la constante α , también llamada difusividad térmica. Pero en un incendio real la transmisión de calor no es el único factor que entra en juego, también hay que parametrizar diferentes factores.

$$T = H \frac{W}{t} \quad \text{Ecuación 3}$$

Para la modelización de la combustión debemos tener en cuenta la temperatura de la celda y la cantidad de combustible W en esa posición. Cuando aumentamos la temperatura de ignición se produce el fuego, quemando el combustible y aumentando la temperatura en esta posición. En la ecuación 3 podemos observar como la temperatura aportada es calculada mediante la cantidad de combustible en esa posición W partido por el tiempo t que va a estar ardiendo y multiplicado por la constante H que indica la cantidad de temperatura emitida por el material ardiendo. Una vez la celda se encuentre sin combustible se considera que las llamas se han extinguido y la celda dejara de aportar calor.

Si tenemos en cuenta únicamente estas dos ecuaciones nos encontraríamos frente a un modelo idealista que solo puede aumentar de temperatura. En el mundo real no todo el calor generado se transmite, sino que existe una pérdida en el ambiente. Este factor de pérdida de calor al ambiente es producido por la dispersión del calor por el medio del aire,

reduciendo la temperatura que se trasmite en la ecuación del calor. En la ecuación 4 podemos ver como la variación de temperatura a lo largo del tiempo es la diferencia entre la temperatura ambiente T_{amb} y la temperatura en ese punto T dividido por la constante de perdida de calor t_{hl} .

$$\frac{\partial T}{\partial t} = \frac{T_{amb} - T}{t_{hl}} \quad \text{Ecuación 4}$$

Por último, tenemos la modelización del viento. Influye principalmente en la dirección y velocidad por el cual el incendio se esparce por el terreno. Las llamas en un punto experimentan un pequeño decremento de temperatura en la dirección opuesta a la velocidad del viento v , dificultando la expansión de las llamas en esa dirección, como se observa en la ecuación 5.

$$T(h, x_0) \approx T(0, x_0) - hv \frac{\partial T}{\partial x}(0, x_0) \quad \text{Ecuación 5}$$

Con la implementación de estas ecuaciones y la división del terreno se podrá calcular la temperatura y la cantidad de combustible de cada una de las celdas en el instante t y se actualizarán sus valores para cada $t + 1$. En caso de que las temperaturas superasen el umbral de fuego, la celda empezara a arder hasta quedarse sin combustible.

Como hemos comentado anteriormente, dependiendo del combustible del terreno el incendio, la expansión de las llamas se desarrolla de una manera distinta, en zonas frondosas las llamas perduran durante grandes instancias de tiempo desprendiendo enormes cantidades de temperatura mientras que en zonas con menos vegetación las llamas permanecen menos tiempo debido a la escasez de combustible de la zona.

Para determinar este mapeo de combustible del terreno se ha utilizado una función de Ruido Perlin. La función de ruido Perlin es una función matemática que aporta valores pseudoaleatorios a lo largo de un espacio utilizando la interpolación de gradientes. En la Figura 3.2 podemos ver un ejemplo de ruido Perlin con valores comprendidos entre 0 y 1 en donde los valores más bajos son representados con el color negro y los valores más alto con el color blanco.

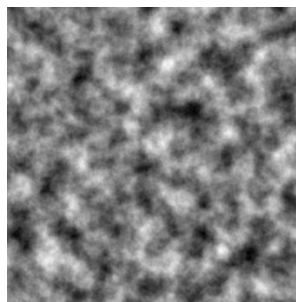


Figura 3.2: Ejemplo Ruido Perlin.

Con todas estas funciones modelizadas se ha generado un *script* en Unity donde asignándola a un terreno obtenemos una simulación de la propagación en un incendio forestal. Una vez obtenido este incendio se llevó a cabo un proceso de ajuste de los diferentes valores con el objetivo de obtener una simulación realista que se ajustase tanto al terreno como a los diferentes componentes del sistema.

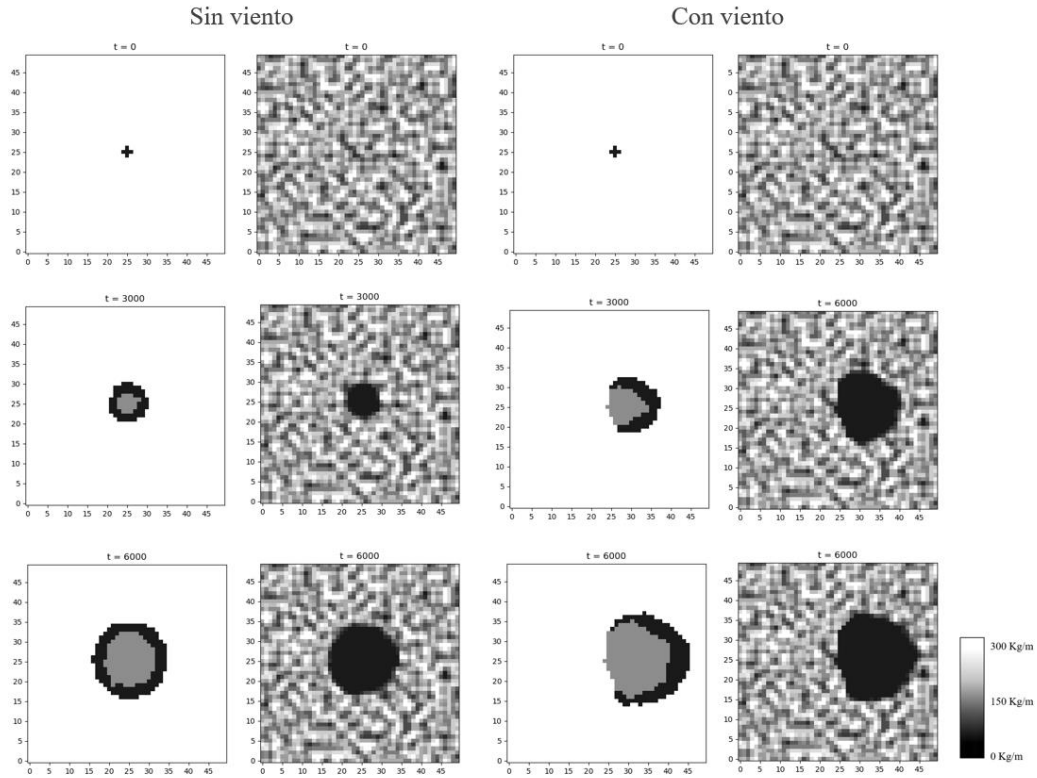


Figura 3.3: Ejemplo evolución incendio.

En la Figura 3.3 podemos observar la evolución de dos incendios utilizando la modelación del incendio forestal. En las columnas impares muestran el desarrollo del incendio en el mapa del terreno discretizado, mostrando en blanco las casillas que se encuentran en estado normal frente a las casillas negras que se encuentran en llamas. En las columnas pares podemos ver la evolución del combustible de la zona a lo largo del tiempo siendo el color más oscuro la identificación de la inexistencia de combustible, debido a que las llamas ya lo han consumido.

Ambos incendios están provocados en la zona central del terreno siendo las dos primeras columnas un incendio sin viento, provocando que se expanda en todas las direcciones con la misma intensidad. mientras que las dos columnas siguientes muestran la evolución de un incendio con un ligero viento en dirección este.

Como se puede observar a medida que transcurre el tiempo el incendio se expande en todas las direcciones, consumiendo el combustible de las celdas por las que atraviesa. En el caso del viento, las llamas se expanden a mayor velocidad en la dirección del viento mientras que las llamas se encuentran con mayor resistencia en sentido contrario, provocando una lenta expansión en esa dirección.

3.5 Drones

Como se ha comentado en la sección 2, existen una gran variedad de vehículos utilizados en tareas de detección y monitorización como drones, aeronaves de ala fija o helicópteros. Pero la aplicación de drones se está extendiendo rápidamente debido a las grandes ventajas que presenta respecto a otros vehículos más convencionales. La mayoría de estas ventajas vienen debido al reducido tamaño de la aeronave permitiéndole un fácil método de transporte en las áreas de utilización, su rápido despliegue, la posibilidad de maniobras en situaciones de difícil acceso y su bajo coste.

Un helicóptero para tareas de incendio puede suponer un coste de 4.000 €/h y su precio aumenta considerablemente más si tenemos en cuenta la adquisición de la propia aeronave. Su coste elevado implica una gran restricción en el número de vehículos que se pueden llegar a utilizar. Por otro lado, el coste de un dron comercial ronda los 500€ y su coste de vuelo es muy reducido. Esto es de gran importancia debido a la naturaleza de estas tareas, donde la integridad de la nave se encuentra en constante peligro.

Debido a su reducido coste y a su fácil transporte a las zonas afectadas por llamas este tipo de aeronaves está creciendo en popularidad en tareas de detección y monitorización. Por estas razones se ha optado por la utilización de drones dentro del simulador, en el cual se les ha añadido un modelo similar a los drones que se encuentran actualmente comercializando.



Figura 3.4: Modelo 3D de un dron.

El movimiento de un dron se produce gracias al gran empuje realizado por sus 4 hélices que le permite sostenerse en el aire. Con la información aportada por los sensores de la aeronave como la inclinación, las revoluciones de las hélices y el velocímetro permiten al software de control variar la rotación y velocidad de las hélices permitiéndole maniobrar. Los últimos modelos de estos últimos años permiten, además del control remoto, sistemas de suspensión en el aire y su navegación programada a través de direcciones o con una serie de coordenadas que debe seguir (*waypoints*).

Una vez creado el entorno de simulación dentro de Unity nos ponemos a simular las aeronaves dentro de la herramienta. Se ha importado el modelo 3D, Figura 3.4, de un dron y se ha codificado un *script* para simular el comportamiento de un dron real. Este *script* será el encargado de simular las características como el tamaño, el peso y la velocidad de la aeronave.

Otro parámetro que se necesita tener en cuenta es la autonomía de la batería, la cual en los modelos actuales suele oscilar entre los 25 y 35. Estos valores pueden variar dependiendo del modelo y de la velocidad de la aeronave, a mayor velocidad mayor consumo. Para su recarga se ha decidido simular estaciones de carga que se situaran en los extremos del entorno de simulación. El dron que entre en contacto con la estación de carga, le aumentara la carga de la batería de manera progresiva. En una situación real una carga de una batería de ese tamaño puede llevar horas, esta estación de carga sería más bien una

simulación del intercambio de una batería de un dron por otra que se encuentre totalmente cargada.

El control del dron se maneja con un script que controla tanto la dirección y velocidad de la aeronave como el consumo de la batería. El consumo de batería es ajustado dependiendo la velocidad de la aeronave y adaptando a la simulación del incendio para obtener una autonomía de alrededor de 30min.

La aeronave será controlada con un vector velocidad 2D. Se ha optado por este método del control del dron frente al control de las velocidades de cada hélice con el objetivo de simplificar el problema a resolver. Además, actualmente existe en el mercado software que ajusta la velocidad de las diferentes hélices de la aeronave de manera automática [25, 26].

3.5.1 Sensores

Para que el agente pueda interactuar con su entorno y tomar decisiones es necesario implementar diferentes sensores para que pueda percibir el estado del incendio. Se ha optado por usar una serie de sensores reales que se han replicado dentro del entorno de simulación. Estos sensores serán implementados en todas las aeronaves. Son los siguientes:

- **Modulo GPS:** Dispositivo acoplado a la aeronave que es utilizado para determinar la posición de las aeronaves en cada momento. Para simular este sensor utilizaremos la posición x y z del dron en el campo de entrenamiento.
- **Sensores de proximidad:** Existen numerosos sensores de distancia para drones en el mercado, los más comunes son los sensores LIDAR. Estos sensores funcionan emitiendo pulsos de luz que rebotan en las superficies. Dependiendo del tiempo que haya transcurrido desde que se emite hasta que se recibe el pulso se puede determinar la distancia a los diferentes objetos del entorno. En la actualidad los drones de alta gama ya presentan este tipo de tecnologías capaces de mapear y detectar objetos a su alrededor con un rango de hasta 20m [27]. En entorno de Unity cuenta con un componente llamado *rayCast* que consiste en un rayo de distancia finita el cual indica si encuentra un objeto en la dirección en la que se encuentra. Para la simulación de estos sensores se han situado varios *rayCasts* alrededor de la aeronave los cuales son utilizados para determinar si hay un obstáculo en sus alrededores.



Figura 3.5: Sensor UST-20LN.

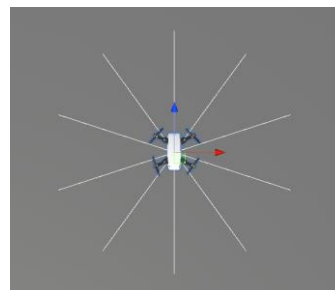


Figura 3.6: Raycast en Unity.

- **Cámara térmica:** Cámara acoplada a la aeronave orientada hacia abajo, con el objetivo que pueda detectar incendios. Existen diferentes publicaciones donde se

demuestra el uso de este tipo de cámaras para la detección [28]. En el mercado existe gran variedad de modelos de cámaras térmicas y algunos drones de uso industrial ya cuentan con esta tecnología implementada. Para su simulación en Unity se ha decidido realizar una simplificación aportando al dron que pueda detectar la temperatura de la celda en la que se encuentra y de las celdas vecinas. Si la celda supera cierto umbral se considera que la celda se encuentra en llamas. En la Figura 3.6 podemos observar un ejemplo del área de detección de una aeronave en la simulación.

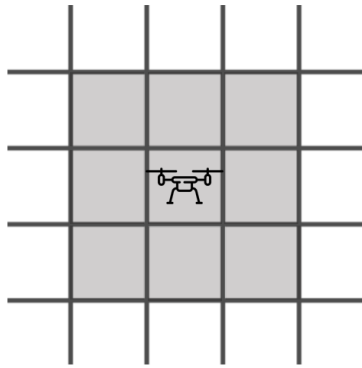


Figura 3.6: Esquema de las celdas detectadas por el dron.

- **Giroscopio y velocímetro:** Sensores integrados en el controlador del vuelo de la aeronave capaces de determinar la orientación, rotación y velocidad en cada momento. En Unity estos sensores se han simulado obteniendo directamente los valores de orientación, rotación y velocidad del dron *GameObject* (objeto simulación en Unity).

Una vez realizado el entorno de simulación y la reproducción de los drones en la herramienta Unity se dispone a explicar el diseño y desarrollo de la arquitectura de los comportamientos.

3.5.2 Arquitectura comportamientos

La habilidad del sistema global está determinada por los comportamientos que lo componen. Para la implementación de estos comportamientos primero fue necesario definir las diferentes tareas que debería realizar la aeronave en una tarea de monitorización. El sistema está formado por una tarea principal encargada sobrevolar el incendio obteniendo información y dos tareas para mantener la integridad de la nave el mayor tiempo posible. Se han decidido 3 comportamientos principales:

- **Bordear incendio:** Encargado de sobrevolar la zona del incendio afectada por las llamas obteniendo la mayor información del incendio posible.
- **Recargar batería:** Encargado de abandonar la zona del incendio para sobrevolar hacia un punto de carga. Este mismo comportamiento será también encargado de sobrevolar de vuelta a la zona afectada una vez tenga la batería recargada.
- **Esquivar Obstáculos:** Encargado de prolongar la duración del vuelo esquivando cualquier obstáculo presente en el terreno que sobrevuela como arboles de gran altura, tendidos eléctricos u otras aeronaves de la zona.

El sistema recibirá como entrada aquella información captada por la aeronave en el entorno de simulación. Para simplificar el entrenamiento de los comportamientos, cada uno recibirá únicamente como entrada la información necesaria para la toma de decisiones, así pues, por ejemplo, la cantidad de energía de la batería no será enviada al módulo de esquivar obstáculos. Los comportamientos tendrán una función diferente y son entrenados de manera independiente obteniendo como salida un vector velocidad de dos dimensiones. Los datos de entrada al sistema son:

- Posición del dron.
- Posición de los drones compañeros.
- Velocidad del dron.
- Posición del ultimo fuego detectado.
- Posición de la estación de carga más cercana.
- Cantidad de batería restante de la aeronave.
- Distancia (si encuentra) al obstáculo de los sensores de proximidad.

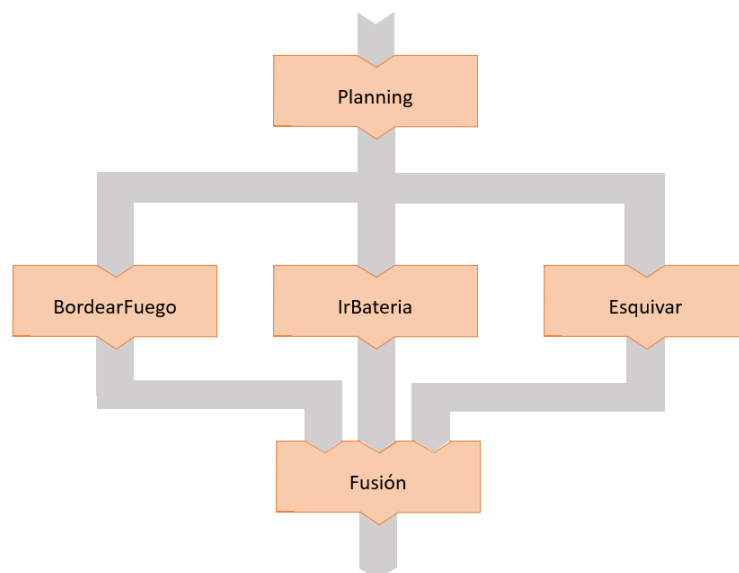


Figura 3.7: Esquema arquitectura del sistema

En la Figura 3.7 podemos ver un esquema simplificado del diseño de la arquitectura basada en comportamientos. En ella el módulo de planificación obtiene la totalidad de los datos de entrada al sistema y envía a cada comportamiento aquellos que necesite utilizar. Cada uno de los comportamientos, dado los datos de entrada, decide la mejor decisión que debe tomar y aporta como salida un vector velocidad. Y por último el módulo de fusión obtiene las diferentes salidas de los comportamientos y decide la salida final del sistema a utilizar que es pasado posteriormente al control del dron para que efectúe la acción.

Este tipo de arquitecturas presentan una gran escalabilidad dada la facilidad que presenta a poder incrementar el número de comportamientos del sistema. A continuación, se explicarán más en profundidad los diferentes módulos que forman el sistema empezando con cada uno de los comportamientos.

3.6 Comportamientos

En este apartado se expondrán los comportamientos implementados en el sistema explicando la funcionalidad que persiguen y su desarrollo.

3.6.1 Bordear Fuego

Como se ha comentado anteriormente, la expansión de las llamas es una acción muy aleatoria y peligrosa que puede variar en pocos minutos. El objetivo de este comportamiento es conseguir una trayectoria de vuelo para el dron con el objetivo de medir la expansión del incendio y detectar posibles focos alejados del principal. Con el comportamiento activado, la aeronave sobrevolará la zona del incendio, actualizando el mapa de conocimiento interno y consiguiendo que se asemeje lo más posible a la expansión del incendio real. Esta información será facilitada a los profesionales en la lucha contra las llamas para mantenerles informados de la situación actual de las llamas y para poder realizar futuras tomas de decisiones críticas para su tera. Este comportamiento se podría considerar el principal del sistema.

Debido a las numerosas variantes externas que afectan a la expansión del incendio como el viento, el combustible, el terreno etc., el incendio puede tomar numerosas formas y velocidades de expansión. Esto convierte la tarea de bordear la frontera de las llamas una tarea difícil de codificar.

Se ha decidido el uso de aprendizaje con refuerzo con redes neuronales para lograr la conducta deseada. Para su entrenamiento se ha utilizado el simulador de incendios provocando un fuego inicial en la zona central del terreno con dos drones agentes, con el objetivo de trabajar en equipo para conseguir la mayor información del área afectada por el incendio.

Dadas las limitaciones de los sensores, cada vehículo no podrá disponer del estado total del incendio, sino que obtendrá información únicamente de las zonas que sobrevuele. A los agentes se les dado los siguientes sensores para tener información de sus alrededores:

- Posición del agente con coordenadas x y z.
- 6 sensores de proximidad alrededor del dron.
- Posición del fuego más cercano detectado.
- El ángulo formado entre la dirección de la aeronave y el incendio más cercano.

Esta información obtenida por los sensores será las entradas de datos de la red neuronal que será entrenada. Se ha utilizado una arquitectura de 3 capas densamente conectadas con 512 neuronas cada una.

Durante el entrenamiento, el dron recibirá una recompensa al sobrevolar una celda que se creía que no estaba ardiendo y al comprobar que se encuentra en llamas. Esta recompensa fomenta al dron a sobrevolar la frontera del incendio obteniendo la mayor información posible. El agente debe cooperar con el resto de los agentes en el campo de entrenamiento con el objetivo de abarcar el mayor terreno posible, así pues, el agente no es recompensado por las celdas que previamente hayan sido visitadas por otro dron. De esta manera se fomenta que los agentes abarquen áreas que no hayan sido sobrevoladas anteriormente para maximizar la recompensa obtenida. Además, para asegurar la distancia

de separación se les aplica una pequeña penalización al acercarse demasiado al resto de aeronaves. Por último, para garantizar la integridad de la aeronave se les aplica una pequeña penalización cuando se sitúan exactamente encima de las llamas. Esa posición puede alcanzar enormes temperaturas que dañarían los sensores y los componentes del robot.

3.6.2 Ir batería

En un despliegue real frente a un incendio forestal, los bomberos suelen organizarse en una zona segura próxima al incendio donde poder organizarse y gestionar las tareas de extinción. Estas zonas son perfectas para el despliegue de los drones y para las acciones de carga/intercambio de baterías.

Como se ha comentado en la Sección 3.5, al dron de simulación se le ha aportado de una batería la cual se va reduciendo a medida que pasa el tiempo y que realiza movimientos. En el caso que el robot se quede sin batería en mitad de una operación, caería en las zonas de las llamas donde se perdería todo el hardware. La aeronave debe ser consciente de la cantidad de carga restante con la que se encuentra para que pueda alejarse de la zona de operación hacia zonas seguras donde se pueda manipular el robot sin riesgo alguno. El objetivo de este comportamiento es obtener una ligera gestión de la batería, con el objetivo de que el agente obtenga un comportamiento de autoconservación haciendo que el agente se aproxime a las estaciones de carga situadas en zonas más alejadas cuando el agente se encuentre con batería reducida.

Para este comportamiento se han usado estaciones de carga situadas en los extremos del campo y un fuego fijo situado en la zona central. Se pretende con este comportamiento que el dron sea capaz de mantenerse quieto junto al juego hasta que obtiene un valor de batería lo suficientemente bajo para ir a recargarse. La información utilizada por este comportamiento es:

- La posición de la aeronave.
- Carga de la batería restante.
- Posición a la estación más cercana.
- Posición de fuego más cercana.

Debido a la simplicidad de este comportamiento se ha decidido utilizar un sistema determinista de estados que se encargue de gestionar la conducta. El comportamiento se activará cuando el dron se encuentre con un porcentaje bajo de batería y se dirigirá a la estación disponible más cercana donde se recargará la batería. Una vez se encuentre con la batería recargada se desplazará hacia la zona del incendio. El comportamiento se mantendrá en estado inactivo cuando tenga un porcentaje de batería superior al 15%. En la figura 3.7 podemos ver el esquema simplificado.

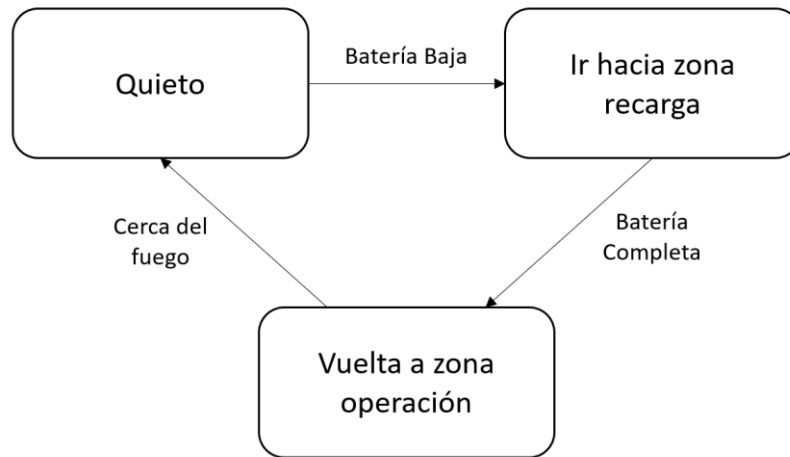


Figura 3.8: Estructura determinista del comportamiento.

3.6.3 Evitar obstáculos

Los terrenos silvestres donde se producen la mayoría de los incendios forestales son terrenos con difícil acceso y la navegación de los drones se puede dificultar debido a la aparición de obstáculos como árboles, tendidos eléctricos u otras aeronaves que sobrevuelan la zona. Cualquier colisión con alguno de estos obstáculos supondría la precipitación de la aeronave y la finalización de la operación de monitorización por su parte. Además, en caso de su precipitación supondría una complicada tarea de recuperación de la aeronave o incluso el recambio completo del robot con los gastos que supondría. Es por eso por lo que es necesario tener un sistema que mantenga la integridad del dron durante el vuelo.

El objetivo de este comportamiento es ser capaz de realizar un cambio de dirección o maniobrar en caso de acercarse demasiado a un obstáculo que pueda dañar e incluso acabar con el vuelo de la aeronave. Debido a la complicada codificación de este comportamiento se ha decidido utilizar la misma estrategia que el comportamiento de bordear incendio y utilizar entrenamiento por refuerzo para su desarrollo.

En su entrenamiento se ha dispuesto un terreno con diferentes modelos 3D simulando aquellos que se pueden encontrar en una zona forestal como arboles demasiado altos, postes eléctricos o incluso otras aeronaves. Como entrada de datos para las redes neuronales se les ha facilitado:

- La posición de la aeronave.
- 6 sensores de proximidad alrededor del dron.

Para el entrenamiento se aproximarán distintos obstáculos hacia el agente. El agente recibirá una ligera recompensa en cada instante de tiempo que se encuentre quieto sin realizar ninguna acción. Cuando entre en colisión con alguno de estos obstáculos el agente será penalizado y acabara el episodio del entrenamiento. Con estas reglas se pretende que el agente se encuentre inactivo el mayor tiempo posible y que se realice una maniobra únicamente cuando detecte un obstáculo lo suficientemente cerca. Se ha utilizado la misma arquitectura que en el comportamiento de bordear incendio de 2 capas densas con 512 neuronas cada una.

3.6.4 Planificador y módulo de fusión

Una vez explicado todos los comportamientos tenemos que plantear un sistema que englobe cada uno de ellos y decida cual utilizar en cada momento. Una vez entrenado cada comportamiento de manera independiente podemos obtener su modelo e incorporarlo dentro del agente final.

Dado que cada comportamiento decide una acción a tomar dependiendo de la información que recibe, es necesario tener un criterio de decisión. Dentro de los distintos tipos de mecanismos se ha optado por un mecanismo de arbitraje de *State-based-mechanism*, en donde un módulo extra denominado planificación decide el valor del estímulo que recibe cada comportamiento. El planificador recibe la entrada de los sensores del sistema y determina el estado en el que se encuentra. Una vez determinado el estado decide el valor del estímulo para cada comportamiento en función al grado de relevancia del comportamiento en ese momento.

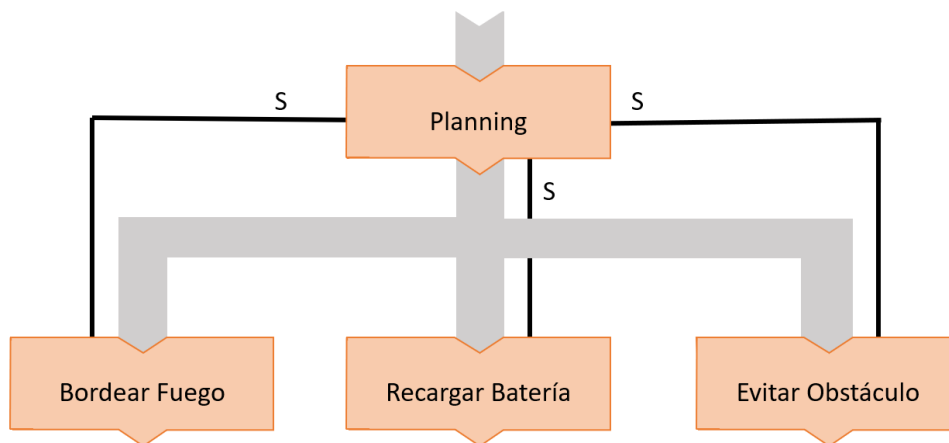


Figura 3.9: Esquema simplificado del módulo de planificación.

Para su desarrollo se ha optado por usar un sistema determinista. Este sistema típicamente aporta un mayor estímulo al comportamiento de bordear incendio, con el objetivo de conseguir la mayor cantidad de información del incendio posible. A medida que transcurre el tiempo y el dron va reduciendo su batería se empezará a aportar mayor estímulo al comportamiento de recarga. El estímulo del comportamiento de evitar obstáculos normalmente será un valor nulo y aumenta a medida que detecta un obstáculo en las inmediaciones.

Por último, tenemos el módulo de fusión el cual es el encargado de recibir el vector velocidad salida y el estímulo de cada comportamiento y decidir la salida del sistema global. Se ha utilizado un mecanismo *Winner-takes-all-mechanism*, en donde la salida resultante es aquella que presenta un mayor estímulo. Esto representa el uso de la salida del comportamiento que mejor se adapta al estado del entorno en ese momento. En este mecanismo, el resto de comportamientos no presentan ninguna influencia a la salida resultante. Una vez decidida la salida del sistema final, este vector velocidad es enviado al *script* que gestiona el movimiento del dron para que se refleje dentro de la simulación.

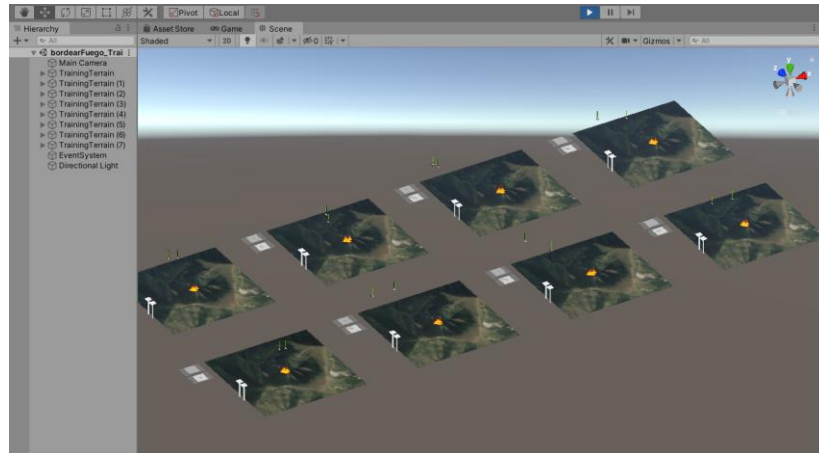


Figura 3.10: Entorno de Unity entrenando el comportamiento de *Bordear Fuego*.

Una vez definidos y desarrollados los diferentes módulos y componentes de la simulación, se llevó a cabo el entrenamiento de los diferentes comportamientos. Al tener un entorno virtual completamente editable se pudo incrementar el paso del tiempo dentro de la simulación. Asimismo, utilizando una ventaja de la librería ML-Agents se pudo poner varios campos de entrenamiento para aportar más información al modelo durante el aprendizaje. Estas ventajas supusieron una enorme reducción en el tiempo de entrenamiento de los diferentes comportamientos. En la Figura 3.9 podemos observar el entorno de simulación de Unity con 8 campos de entrenamiento entrenando de manera simultánea el comportamiento de *Bordear Fuego*.

4 Pruebas y resultados

4.1 Introducción

Tras la etapa de desarrollo se llevaron a cabo diferentes pruebas con el objetivo de medir y valorar la efectividad tanto de los comportamientos por separado como de la eficacia del sistema global. En esta sección mostraremos los resultados obtenidos de los dos comportamientos desarrollados con aprendizaje por refuerzo y los resultados del sistema global en un caso simulado. Adicionalmente se llevaron a cabo varias pruebas variando el número de drones desplegados y variando la silueta del incendio para comprobar la generalización del conocimiento aprendido.

4.2 Resultados de cada comportamiento

Cada comportamiento ha necesitado un ajuste tanto de los hiperparámetros del algoritmo utilizado, como de las recompensas y penalizaciones a los agentes según las acciones que realicen. El ajuste de estos parámetros es crítico para el resultado porque determina tanto la conducta final del comportamiento como la velocidad del aprendizaje.

La metodología utilizada consistió en realizar unos ajustes de manera manual y tras la etapa de entrenamiento se analizaba tanto las gráficas de las recompensas acumuladas a lo largo del tiempo como la actuación del comportamiento en diferentes situaciones. Una vez analizadas estas métricas se ajustaron los parámetros y se repitió el proceso hasta obtener una conducta del comportamiento óptimo.

Bordear Fuego:

Los principales problemas encontrados durante el desarrollo han sido las numerosas modificaciones que se han tenido que hacer para ajustar las cantidades de recompensa y penalización que obtenía el agente cuando realiza una determinada acción. Recompensas muy altas en determinadas acciones provocaban las repeticiones de acciones no deseadas y recompensas muy bajas no incentivaban suficiente a los agentes a realizar las acciones. Por eso es necesario encontrar un balance adecuado.

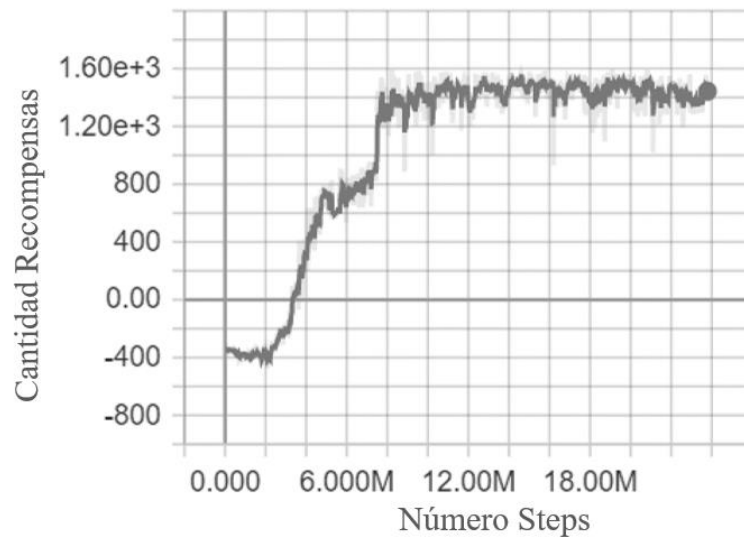


Figura 4.1: Evolución de recompensas a lo largo del tiempo.

En la Figura 4.1 podemos ver la evolución de las recompensas a lo largo de los diferentes *steps* de la simulación. En la librería de Unity un *step* es considerado cada paso de tiempo de la simulación que suele corresponder con cada actualización de fotograma. Para este entrenamiento se utilizó una longitud de episodio máximo de 6000 *steps*. Podemos ver como durante los primeros 2M *steps* los agentes se están familiarizando con el entorno. A partir de los 2M *steps* los agentes aprenden a mantenerse dentro de los márgenes de la zona de entrenamiento y mantenerse alrededor del incendio. A partir de los 8M de *steps* el modelo converge, consiguiendo la máxima recompensa posible y se mantiene constante durante el resto del entrenamiento.

Tras el entrenamiento, los resultados obtenidos fueron bastante adecuados, los agentes fueron capaces de sobrevolar la zona del incendio, manteniendo una distancia de seguridad con las llamas realizando movimientos rotatorios en posiciones opuestas respecto al otro dron que se encuentra en el incendio con el objetivo de maximizar las recompensas.

Un problema encontrado es que al dar a los agentes un área de detección de incendios reducida y fomentar con recompensas a los agentes a no encontrarse sobre las llamas, el agente tiene problemas al actualizar las celdas que se encuentran en la parte interior del anillo de fuego y no detecta de manera correcta cuando se han apagado las llamas.

No es un problema mayor debido a la información relevante del incendio son las celdas que se encuentren en llamas y con la suposición que las celdas que se encuentran en llamas dado un transcurso de tiempo se queda sin combustible y las llamas se extinguen.

Evitar Obstáculo:

Como hemos comentado en la sección anterior, el agente fue recompensado por conservar su posición y mantenerse el mayor tiempo sin chocarse con los obstáculos. Tras su entrenamiento el agente fue capaz de mantenerse sobrevolando la zona hasta el límite de la longitud del episodio. Debido a que el episodio de entrenamiento duraba hasta que la aeronave colisionaba con un obstáculo podemos observar cómo a medida que transcurre el entrenamiento el agente logra esquivar diferentes elementos consiguiendo aumentar el tiempo medio del episodio.

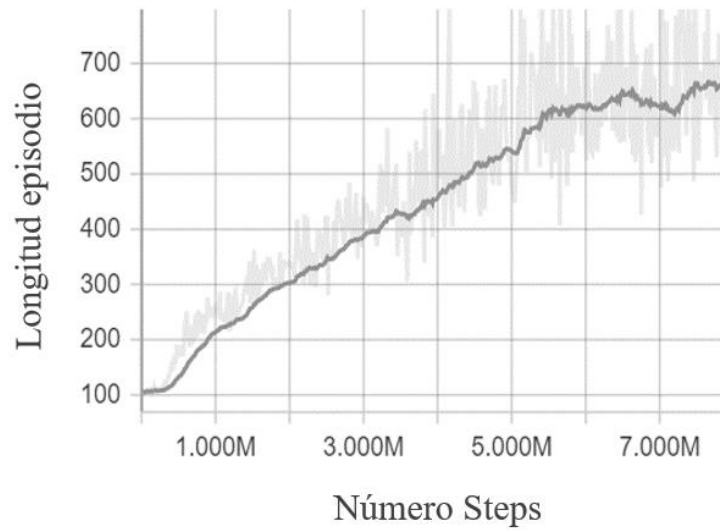


Figura 4.2: Evolución longitud episodio a lo largo del tiempo.

Como podemos observar en la Figura 4.2, los agentes han aprendido a poder esquivar los diferentes obstáculos de manera casi continua hasta alcanzar su máximo donde converge y se mantiene más o menos constante en futuros episodios.

4.3 Resultados sistemas global

Para comprobar el funcionamiento del sistema global se ha diseñado un mapa de creencia, un mapa que se actualiza el estado de cada celda que se encuentran en proximidad de los drones. Este mapa sería el equivalente a la información conocida por los drones y la cual es facilitada a los diferentes profesionales desplegados por la zona. Para los experimentos se situaba un incendio en la posición central del terreno y se han situado 2 aeronaves en las inmediaciones. Durante la simulación se han ido obteniendo información sobre el estado del incendio real, los datos obtenidos por los drones y sus trayectorias para su posterior análisis.

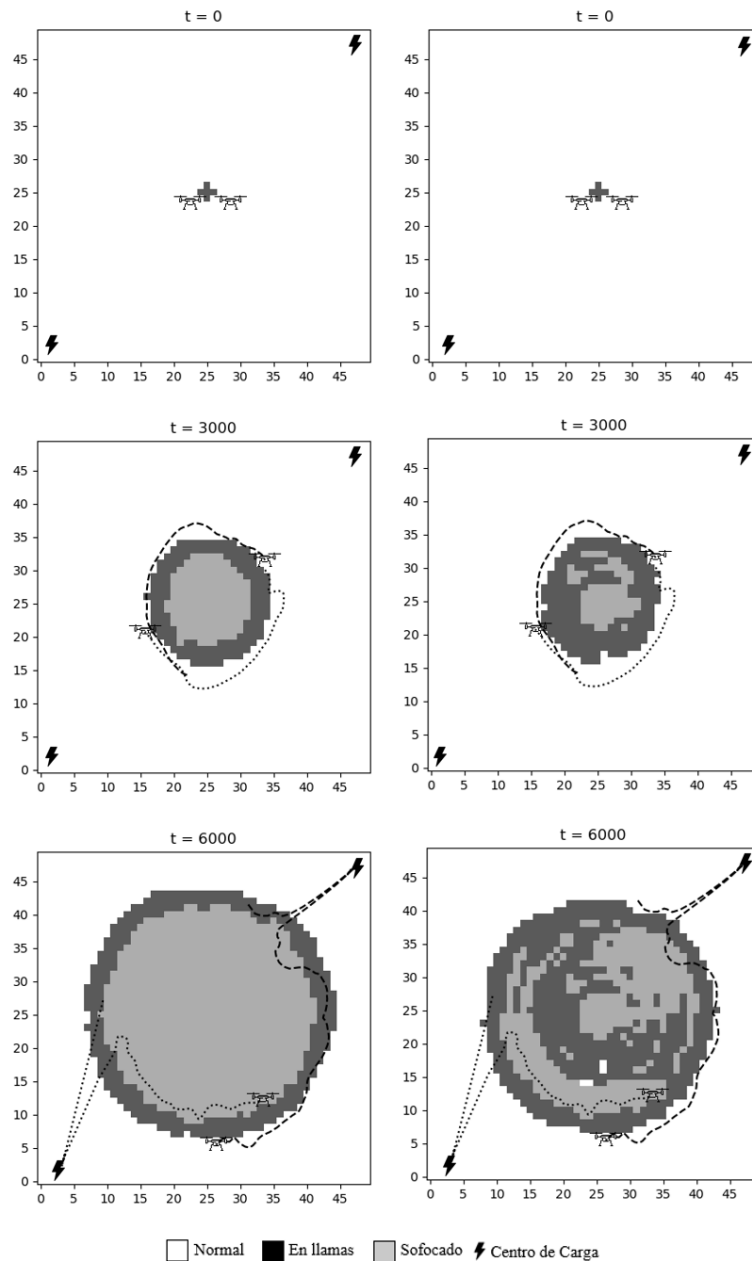


Figura 4.3: Trayectorias de los drones en la simulación.

En la Figura 4.3 podemos observar las trayectorias de los diferentes drones a lo largo del tiempo a la izquierda con el mapa de la expansión y a la derecha con la información captada por las aeronaves. En el instante 0 de tiempo podemos observar como el incendio se acaba de generar en las celdas centrales y se ha dispuesto a las aeronaves alrededor. A medida que la simulación avanza y se expanden las llamas las aeronaves empiezan a sobrevolar la zona alrededor del incendio en sentido horario, situándose en posiciones opuestas entre si con el objetivo de abarcar el mayor terreno posible. Al comparar la expansión del incendio real con los datos obtenidos por los drones podemos ver discrepancias en las zonas internas al incendio. Esto es debido a las restricciones impuestas a los agentes durante el entrenamiento de volar sobre las llamas, para conservar su integridad. De esta manera los agentes sobrevuelan la zona del incendio por la parte exterior y en ocasiones no llegan a detectar el estado de las celdas que se sitúan más en el interior. Esto no se considera un problema debido a que el objetivo del sistema es detectar la expansión del incendio. Las

celdas situadas en el interior del incendio son irrelevantes para los operarios que se encuentran en las tareas de extinción. Estas celdas se pueden considerar que pasado un determinado tiempo se consume todo el combustible y se apagan.

Con el objetivo de comprobar el grado de fiabilidad de aportado por el sistema se compara el porcentaje de fiabilidad entre el mapa de creencia y el área afectada a lo largo del tiempo. Como hemos comentado las celdas situadas en las zonas internas del incendio no aportan mucha información y es por eso por lo que se ha calculado el porcentaje de fiabilidad únicamente con las celdas fronterizas del incendio. Es por eso por lo que contaremos las celdas del incendio real en llamas y se comprobara el estado capturado por los drones.

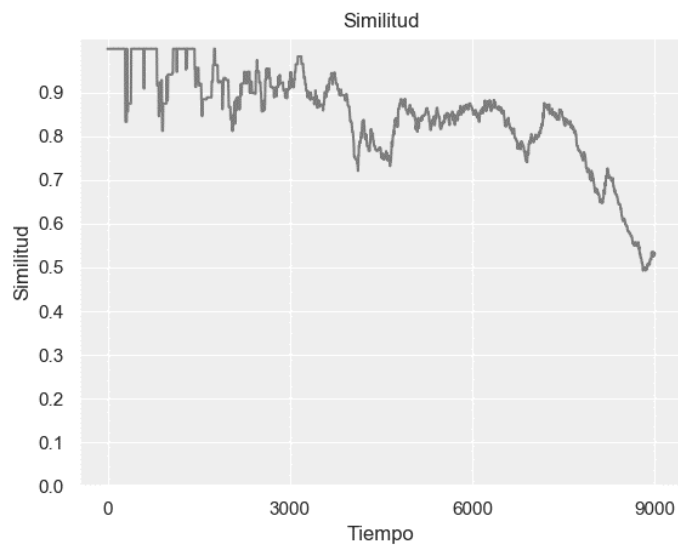


Figura 4.4: Porcentaje de similitud del incendio real por el detectado por los drones.

En la Figura 4.4 tenemos representado el porcentaje de similitud entre el incendio real y la información obtenida de los drones. En los primeros instantes del tiempo las aeronaves tienen un gran porcentaje de similitud, aunque presenta mucha variabilidad que dura muy pocos instantes. Esto es debido a que al considerar únicamente las celdas en llamas y al tener un área pequeña en el incendio en donde las pocas celdas tienen mucho peso en el porcentaje calculado. Al alcanzar un instante de tiempo entre 3.000 y 6.000, en donde tenemos un área del incendio ya considerable podemos ver como se estabiliza la fiabilidad entre un 80% y 90%, con excepción de algunos pequeños valles donde se reduce al 75% debido a la necesidad de las aeronaves de ir a la estación para poder recargar las baterías. Alrededor del instante de tiempo de 8.000 podemos observar como la fiabilidad baja considerablemente. Esto es debido a que, a estas alturas de la simulación, las llamas se han extendido notablemente forzando a los drones a tener que sobrevolar áreas mayores. Este problema es fácilmente solucionable aumentando el número de drones desplegados durante la operación a medida que se incrementa el área afectada por el incendio.

4.1 Otros experimentos

Después de haber probado la eficacia del sistema en una situación real se llevaron a cabo diferentes pruebas para comprobar la eficacia del sistema a pesar de no haber sido entrenado en los escenarios concretos. De esta manera se pretende medir el grado de generalización del sistema. Comprobaremos experimentos variando el número de agentes y variando el contorno del incendio.

4.1.1 Variando el número de drones

El entrenamiento de los comportamientos del sistema se llevó a cabo utilizando únicamente 2 drones. Se llevaron a cabo experimentos utilizando 3 y 4 drones para comprobar el desempeño del sistema. Debido a que las entradas de la red neuronal se diseñaron para permitir únicamente la entrada de la posición de una aeronave, se tuvo que realizar una modificación en el sistema para usar como entrada la posición de la aeronave más cercana.

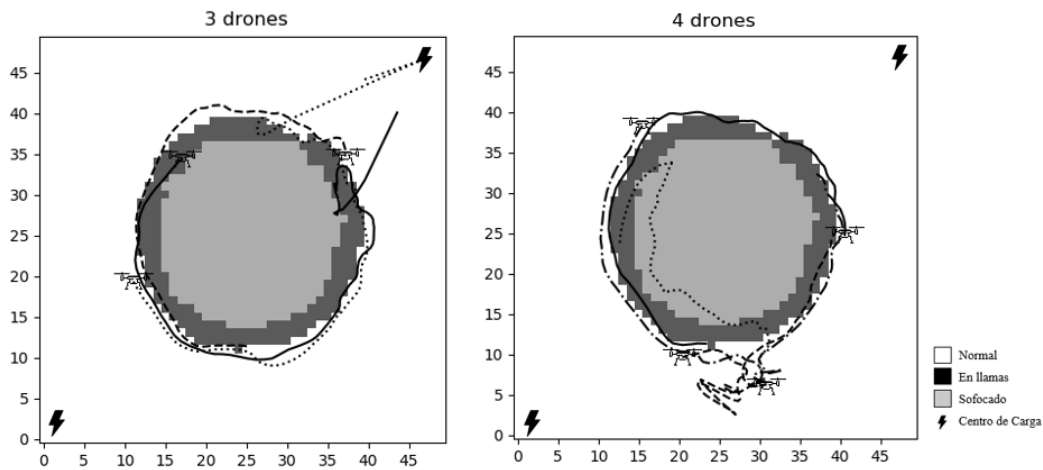


Figura 4.5: Grafica de la simulación con 3 y 4 drones.

En la Figura 4.5 podemos observar las trayectorias realizadas con 3 y con 4 drones. Durante la mayor parte del tiempo los drones fueron capaces de mantener las distancias entre sí y abarcando la mayor distancia posible. Se encontraron algunos problemas en los primeros instantes de la simulación cuando el incendio no se había expandido lo suficiente, los drones al intentar obtener información sobre el incendio se aproximaban lo suficiente y se activaba el comportamiento de esquivar provocando movimientos un poco irregulares. Asimismo, en ambos experimentos se observó que cuando dos aeronaves se aproximaban demasiado entre ellas, generalmente cuando una de ellas regresaba de recargar la batería, tenían movimientos inusuales. Uno de los drones se quedaba a la espera que el otro se alejase hacia la zona opuesta del incendio para continuar con sus movimientos rotatorios.

Estos experimentos demuestran como el sistema puede funcionar aumentando el número de drones. A pesar del correcto funcionamiento en estas situaciones, se ha comprobado movimientos innecesarios que repercute en una obtención de menos información del incendio y un mayor gasto en la batería de las aeronaves. Habría que reforzar la interacción con un mayor número de drones.

4.1.2 Formas Irregulares

En los escenarios reales no todos los incendios van a tener una forma circular, pueden variar con una infinidad de formas. Para que el sistema planteado sea efectivo se debe adaptar a las diferentes formas que puede obtener el incendio. Debido a que el sistema solo ha sido entrenado con formas circulares al analizar formas más complejas podemos medir la habilidad de la red neuronal de generalizar el conocimiento. De esta manera se llevaron a cabo 3 experimentos con formas irregulares:

El primer experimento es un incendio situado de manera central como los experimentados anteriormente, pero esta vez sin ningún tipo de combustible en la parte superior del terreno, puede simular que el incendio se encuentra con un arroyo, una carretera o alguna especie de cortafuegos. De esta manera el incendio se expande hacia la zona sur del terreno formando una forma de semi circunferencia.

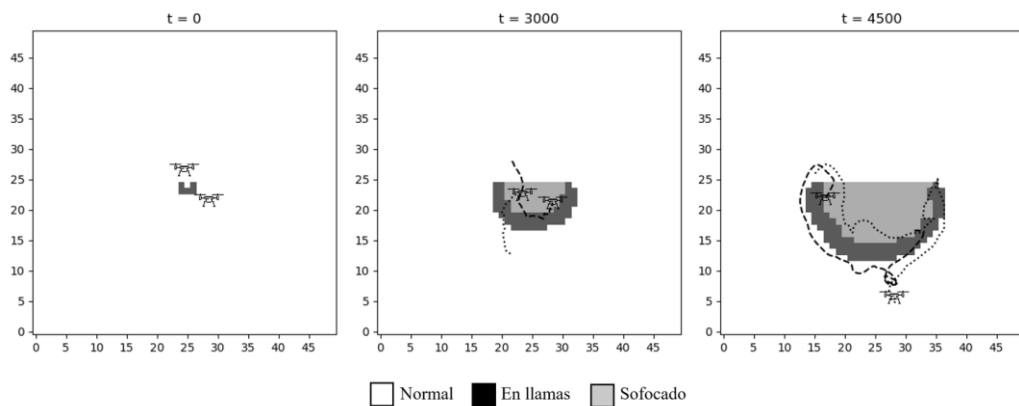


Figura 4.6: Grafica de la evolución del incendio con forma de U.

En la Figura 4.6 se observan las trayectorias descritas por los agentes. Como se puede observar, los agentes fueron capaces de introducirse dentro de la semiesfera y bordear la frontera de las llamas, aunque en alguna situación las trayectorias descritas sobrevolaban las llamas. Esto podría suponer algún problema al poner en peligro la integridad de la nave.

El segundo experimento fue un incendio expandido a lo largo de una línea en vez de en un punto concreto. Se dispuso una dirección de viento en dirección norte.

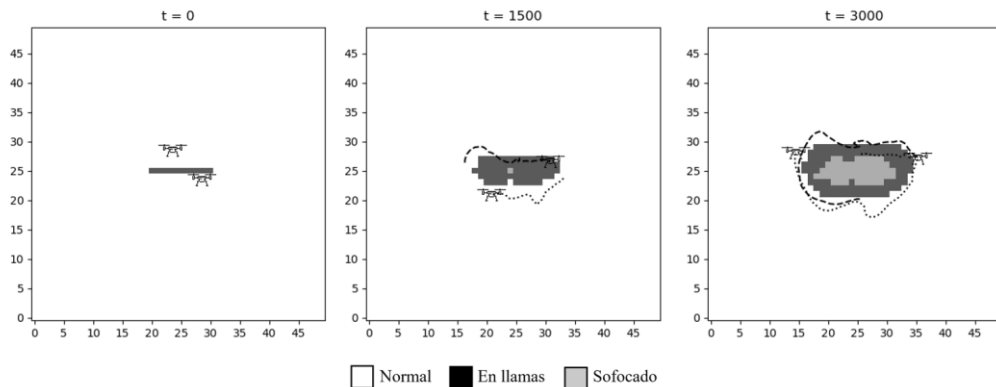


Figura 4.7: Grafica de la evolución de un incendio en forma de línea.

En la figura 4.7 podemos ver el resultado del segundo experimento en donde se simulaba una silueta en línea. Esta silueta se asemeja bastante a la silueta circular con la cual se ha entrenado el sistema, es por esto por lo que se han obtenido unos buenos resultados. Las trayectorias de las aeronaves fueron muy similares a las pruebas iniciales y fueron capaces de sobrevolar la zona sin problema alguno.

Y por último una forma compleja en forma de T, con el objetivo de observar si los agentes son capaces de realizar movimientos más complejos para continuar bordeando el incendio.

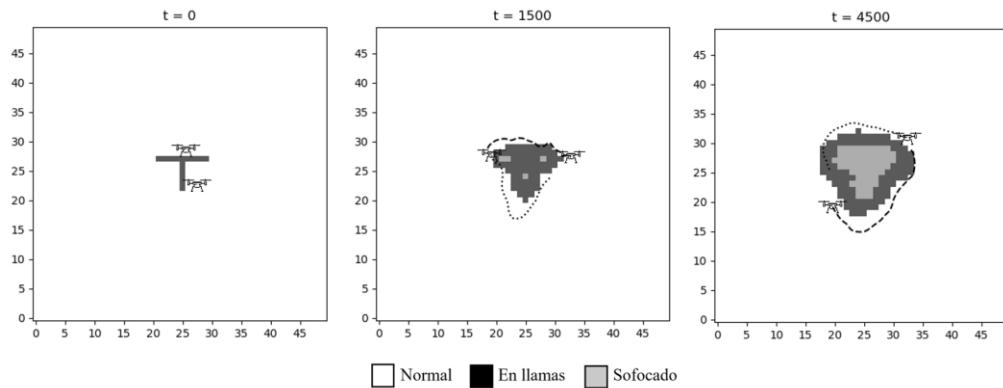


Figura 4.8: Evolución incendio en forma de T.

En la Figura 4.8 se muestran los resultados de las pruebas con las siluetas en forma T. Esta silueta pasada el tiempo suficiente se comporta como la forma circular de un incendio normal, es por eso por lo que los resultados interesantes aparecen en las primeras instancias de la simulación. En los primeros momentos de la simulación podemos observar cómo los agentes realizan movimientos repentinos en los extremos del incendio sin alejarse mucho de la frontera de las llamas. Además, en las esquinas internas de la T los drones se aproximaban lo suficiente para obtener la información de las zonas situadas en la zona central del incendio.

Estos resultados con diferentes formas demuestran cómo se ha conseguido una generalización del conocimiento y un comportamiento útil frente a diferentes siluetas que puede presentar un incendio en diferentes terrenos.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

El resultado principal de este trabajo de fin de grado ha sido el diseño y desarrollo de un sistema de monitorización para aportar información en tiempo real de la evolución de un incendio. En concreto, las principales aportaciones de este trabajo han sido:

- La realización de un modelo de expansión del incendio con alto grado de modificación que ha permitido un entrenamiento más variado para los agentes y la realización de numerosas pruebas para el funcionamiento del sistema global.
- El diseño y creación de 3 comportamientos distintos para el sistema, siendo 2 de ellos realizados con técnicas de aprendizaje por refuerzo para su desarrollo. Y la creación del comportamiento de batería como sistema determinista.
- La implementación de la unión de los 3 comportamientos junto al módulo de planificación y el de fusión.
- La obtención de un 70% y 80% de fiabilidad en incendios no muy extendidos.
- La demostración de la generalización del conocimiento al poder aplicarse a incendios con formas irregulares al haber sido entrenado el sistema con incendios circulares.
- La necesidad del refuerzo del sistema para permitir un mayor número de aeronaves.

5.2 Trabajo futuro

El objetivo del proyecto ha sido estudiar la viabilidad de un sistema de monitorización de un incendio forestal con una arquitectura basada en comportamientos. En consecuencia, abriendo la línea de desarrollo en este campo. Además, al haber desarrollado todos los procesos como la modelización de los incendios, el campo de entrenamiento y la arquitectura de comportamientos permite la profundización en cualquiera de estos elementos y abre numerosas líneas de desarrollo para futuros trabajos.

Una línea de desarrollo que abre sería la implementación del sistema actual en drones reales, lo que implicaría la adaptación del código, la dotación y la implementación de los diferentes sensores.

Otra posible línea de desarrollo podría ser el aumento de parámetros y complejidad del entorno de simulación y modelización del incendio forestal con el objetivo de estudiar diferentes técnicas de actuación y estrategias, además del estudio en diferentes entornos y condiciones climatológicas.

También se abre la línea de desarrollo del aumento de la complejidad del sistema basado en comportamientos planteado al añadir nuevos comportamientos que puedan ser útiles en las operaciones de extinción de las llamas como podría ser escoltar a un pelotón de profesionales para indicarles posibles atrapamientos que se puedan producir por su zona de actuación.

Referencias

- [1] Hernández, L. y cols. (2020). El planeta en llamas. Propuesta ibérica de WWF para la prevención de incendios. WWF España y ANP|WWF. Report. Available Online: https://wwfes.awsassets.panda.org/downloads/wwf_informe_incendios_2020_el_planeta_en_llamas.pdf (accessed on 17 June 2021).
- [2] Greenpeace España (2018) 2017, uno de los peores años de incendios en España. Available Online: <http://archivo-es.greenpeace.org/espana/es/news/2017/Octubre/2017-uno-de-los-peores-anos-de-incendios-en-espana/> (Accessed on 17 June 2021).
- [3] Ministerio para la Transición Ecológica y el Reto Demográfico, Gobierno de España. (2020). Los Incendios Forestales en España: Avance Informativo 2020. Technical report. Available Online: https://www.mapa.Gob.es/es/Desarrollo-rural/estadisticas/avance_1_enero_31_diciembre_2019_tcm30-537398.pdf (Accessed on 17 June 2021).
- [4] Ministerio para la Transición Ecológica y el Reto Demográfico, Gobierno de España. (2020). Los Incendios Forestales en España: Avance Informativo 2019. Technical report. Available Online: https://www.mapa.Gob.es/es/desarrollo-rural/estadisticas/avance_1_enero_31_diciembre_2019_tcm30-537398.pdf (Accessed on 17 June 2021).
- [5] Área de Defensa contra Incendios Forestales, Ministerio de Agricultura, Pesca y Alimentación, Gobierno de España. (2019). Los Incendios Forestales en España: Decenio 2006–2015. Technical report, Available Online: https://www.mapa.gob.es/es/desarrollorural/estadisticas/incendios-decenio-2006-2015_tcm30-511095.pdf (Accessed on 17 June 2021). (s. f.).
- [6] Roldán-Gómez, J. J., González-Gironda, E., & Barrientos, A. (2021). A Survey on Robotic Technologies for Forest Firefighting: Applying Drone Swarms to Improve Firefighters' Efficiency and Safety. *Applied Sciences*, 11(1), 363.
- [7] Drone Hopper - Heavy-duty drone helps fight fires. (s. f.). <https://www.drone-hopper.com/> (accessed on 16 June 2021).
- [8] Pastor Llorens, E., Royo Chic, P., Lopez, J., Barrado Muxí, C., Santamaria Barnadas, E., & Prats Menéndez, X. (2007). Project SKY-EYE: applying UAVs to forest fire fighter support and monitoring. In *UAV 2007: Conference and Exhibition. Unmanned Vehicle Systems International*. Merino, L., Caballero, F., Martínez-de-Dios, J. R., Maza, I., & Ollero, A. (2010). Automatic Forest Fire Monitoring and Measurement using Unmanned Aerial Vehicles. 15.
- [9] Martínez-de Dios, J. R., Merino, L., Caballero, F., & Ollero, A. (2011). Automatic forest-fire measuring using ground stations and unmanned aerial systems. *Sensors*, 11(6), 6328-6353.
- [10] Ambrosia, V. G., Wegener, S. S., Sullivan, D. V., Buechel, S. W., Dunagan, S. E., Brass, J. A., ... & Schoenung, S. M. (2003). Demonstrating UAV-acquired real-time thermal data over fires. *Photogrammetric engineering & remote sensing*, 69(4), 391-402.
- [11] RFSAT - UAV-based photogrammetric 3D modelling and surveillance of forest wildfires. (s. f.). <https://www.rfsat.com/index.php/en/results/technologies/uav-wildfires.html>. (Accessed on 16 June 2021).
- [12] Capitan, J., Mantecon, D., Soriano, P., & Ollero, A. (2007, September). Autonomous perception techniques for urban and industrial fire scenarios. In *2007 IEEE International Workshop on Safety, Security and Rescue Robotics* (pp. 1-6). IEEE. Brewer, M. J., & Clements, C. B. (2020). Meteorological Profiling in the Fire Environment Using UAS. *Fire*, 3(3), 36. <https://doi.org/10.3390/fire3030036>

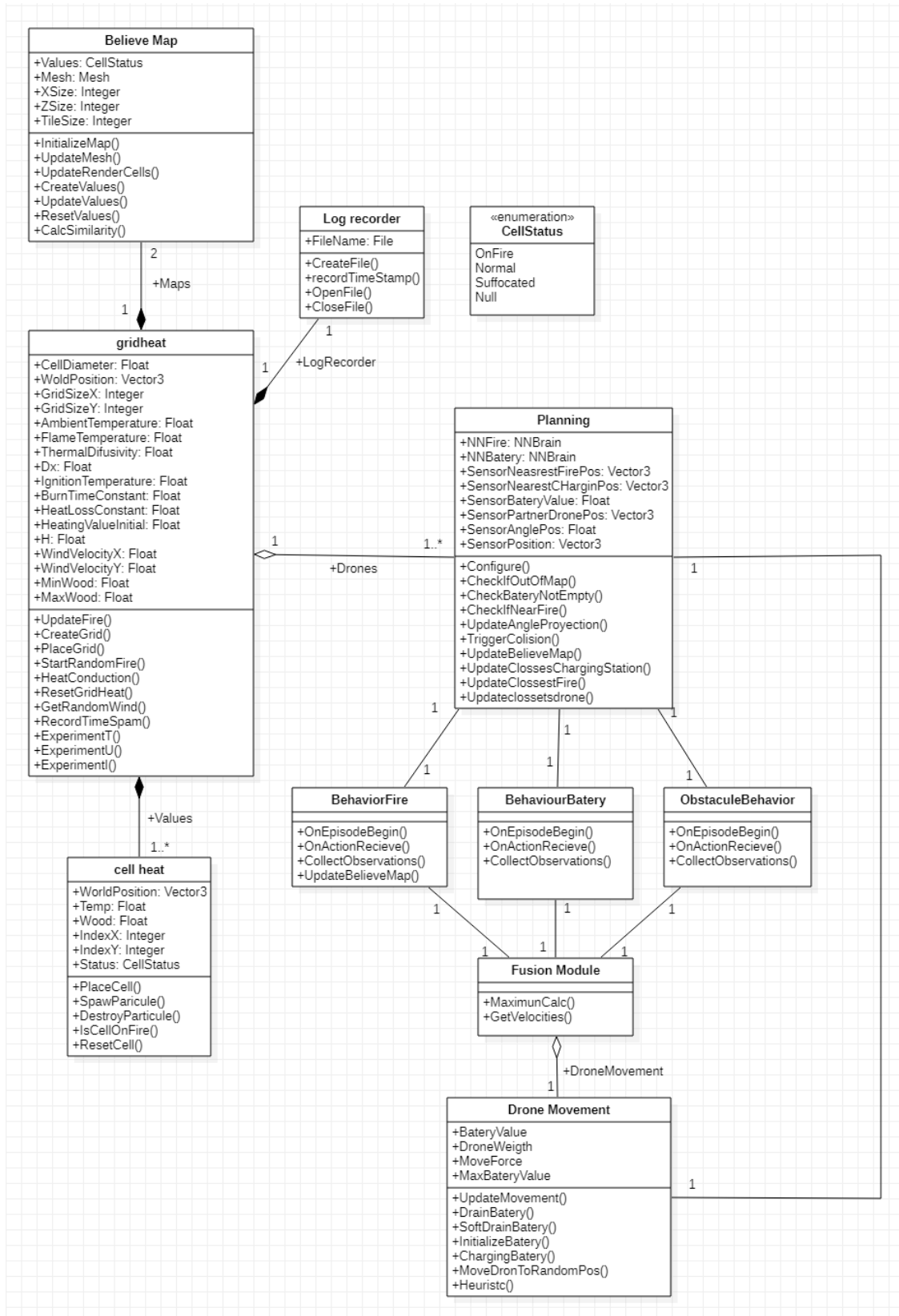
- [13] Ambrosia, V. G., Wegener, S. S., Sullivan, D. V., Buechel, S. W., Dunagan, S. E., Brass, J. A., ... & Schoenung, S. M. (2003). Demonstrating UAV-acquired real-time thermal data over fires. *Photogrammetric engineering & remote sensing*, 69(4), 391-402.
- [14] Ambrosia, V. G., Wegener, S., Zajkowski, T., Sullivan, D. V., Buechel, S., Enomoto, F., ... & Hinkley, E. (2011). The Ikhana unmanned airborne system (UAS) western states fire imaging missions: from concept to reality (2006–2010). *Geocarto International*, 26(2), 85-101.
- [15] VALAQ Patrol – surveillance & security drone features. (s. f.). Available Online: <https://www.valaqpatriot.com/>. (Accessed on 16 June 2021).
- [16] Telefónica, S.A. (2019, June 5). Telefónica successful conducts pilot with drones based on IoT solutions for early fire detection [Press release]. <https://www.telefonica.com/documents/737979/145247426/ndp-dronesantiincendios-eng.pdf/f20b75e8-ceb0-107e-0c2c-ba6cc2a81380?version=1.0>. (s. f.).
- [17] Manzotti, R., & Tagliasco, V. (2005). From behaviour-based robots to motivation-based robots. *Robotics and Autonomous Systems*, 51(2-3), 175-190. Wahde, M. (2012). Introduction to autonomous robots. Lecture Notes from the course Autonomous Agents, Chalmers university of technology.
- [18] Armbrust, C., Braun, T., Föhst, T., Proetzsch, M., Renner, A., Schäfer, B. H., & Berns, K. (2011). RAVON: The robust autonomous vehicle for off-road navigation. In *Using Robots in Hazardous Environments* (pp. 353-396). Woodhead Publishing.
- [19] Wagner, M. D., Apostolopoulos, D., Shillcutt, K., Shamah, B., Simmons, R., & Whittaker, W. (2001, May). The science autonomy system of the Nomad Robot. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation* (Cat. No. 01CH37164) (Vol. 2, pp. 1742-1749). IEEE.
- [20] Proetzsch, M., Luksch, T., & Berns, K. (2010). Development of complex robotic systems using the behavior-based control architecture iB2C. *Robotics and Autonomous Systems*, 58(1), 46-67.
- [21] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press. Kober, J., Bagnell, J. A., & Peters, J. (s. f.). *Reinforcement Learning in Robotics: A Survey*. 38.
- [22] Juliani, A., Berges, V., Teng, E., Cohen, A., Harper, J., Elion, C., Goy, C., Gao, Y., Henry, H., Mattar, M., Lange, D. (2020). Unity: A General Platform for Intelligent Agents. *arXiv preprint arXiv:1809.02627*.
- [23] Sullivan, A. L. (2009). Wildland surface fire spread modelling, 1990–2007. 1: Physical and quasi-physical models. *International Journal of Wildland Fire*, 18(4), 349-368.
- [24] Renardy, M., & Rogers, R. C. (2006). *An introduction to partial differential equations* (Vol. 13). Springer Science & Business Media.
- [25] Bjarre, L. (2019). *Robust Reinforcement Learning for Quadcopter Control*.
- [26] Pham, H. X., La, H. M., Feil-Seifer, D., & Nguyen, L. V. (2018). Autonomous uav navigation using reinforcement learning. *arXiv preprint arXiv:1801.05086*.
- [27] Hokuyo. Hokuyo Sensor UST-20LN Scanning Laser Obstacle Detection Sensor DataSheet. (s. f.). Available Online: https://hokuyo-usa.com/application/files/3916/0857/9123/UST-20LN_Specifications_REV.pdf (accessed on 17 June 2021).
- [28] Martínez-de Dios, J. R., Merino, L., & Ollero, A. (2005). Fire detection using autonomous aerial vehicles with infrared and visual cameras. *IFAC Proceedings Volumes*, 38(1), 660-665.

Glosario

GIF	Grandes Incendios Forestales.
UAV	Unmanned aerial vehicles.
BBR	Behaviour Based Robotics.
GameObject	Componente básico en el entorno Unity.
Asset	Objeto, modelo o archivo importado al entorno Unity.
Waypoints	Conjunto de coordenadas.
Step	intervalo de tiempo en la simulación.

Anexos

A Diagrama de clases



B Material Adicional

Todo el código de este proyecto se encuentra disponible en el repositorio de GitHub para su uso libre:

https://github.com/alexciru/TFG_swarmVSfire

Se añaden también videos de demostración del entrenamiento y de la ejecución del sistema global en el repositorio en el directorio `/videos`.

Para la instalación de proyecto es necesario la instalación de la herramienta Unity junto a la librería ML-Agents. Se recomienda el uso de la versión 2019.3.15f, la cual ha sido la utilizada para el desarrollo. Futuras versiones no han sido probadas. Podrá encontrar la versión correspondiente en el siguiente enlace:

<https://unity3d.com/get-unity/download/archive>

Una vez instalado podrá instalar la librería directamente desde dentro de Unity en la sección de *Window – Packet Manager – add packet – ML-Agent 1.0.7*. Para su correcto funcionamiento es necesario tener instalados las siguientes dependencias en el ordenador. Se recomienda el uso de un gestor de entornos virtuales como puede ser *Anaconda*.

Paquetes Unity	versión
com.unity.ml-agents (C#)	v1.3.0
ml-agents (Python)	v0.19.0
ml-agents-envs (Python)	v0.19.0
gym-unity (Python)	v0.19.0
Communicator (C#/Python)	v1.0.0

Una vez instalado la herramienta Unity como las dependencias puede clonar el repositorio en una ruta y abrirlo con Unity. Una vez clonado podrá abrir las diferentes escenas donde se sitúan los diferentes experimentos realizados.

- **BordearFuego_Training:** Campo de entrenamiento del comportamiento de BordearFuego.
- **EsquivarObstaculos_Training:** Campo de entrenamiento de Esquivar obstáculos
- **SistemaGlobal:** Ejemplo de la ejecución del sistema global en un terreno plano.
- **SampleScene:** Ejemplo de la ejecución del sistema global en un terreno facilitado con relieve.

En cada una de las diferentes escenas la ejecución empezará tras pulsar el botón de Play. Las simulaciones se mostrará los resultados con los modelos pre-entrenados. En caso de querer entrenar un comportamiento deberá ejecutar un comando desde una terminal externa con el comando `mlagents-learn`. Más información en el repositorio del proyecto y en repositorio de ML-Agents.