



Universidad  
Rey Juan Carlos

**Grado en diseño y desarrollo de videojuegos**

**Comportamiento de Personajes**

## **GAME OF DRONES**

**08/01/2021**

### **Profesores**

Dan Casas Guix

Carlos Garre del Olmo

### **Grupo 1**

Juan Manuel Carretero Ávila

Alejandra Casado Ceballos

Pedro Casas Martínez

# Especificación de proyecto

---

## Índice

1.	Descripción general .....	1
2.	Especificación de los personajes .....	2
	Drones .....	2
	Drones de ataque cuerpo a cuerpo y a distancia.....	2
	Reclutador .....	9
	Torreta .....	12
3.	Interacción con el entorno/mundo .....	14
4.	Resultados.....	16
5.	Reparto de tareas .....	23
6.	Webgrafía .....	25

## Índice de figuras

Figura 1. Sistema de utilidad de dron .....	5
Figura 2 Gráficas de las funciones de utilidad.....	5
Figura 3. Máquina de estados de dron ir al punto de espera .....	6
Figura 4. Máquina de estados de dron conquistar o defender una base .....	6
Figura 5. Máquina de estados ataque de dron a distancia .....	7
Figura 6. Máquina de estados ataque de dron cuerpo a cuerpo .....	7
Figura 7. Máquina de estados recluta.....	10
Figura 8. Máquina de estados torreta .....	13
Figura 9. Máquina de estados generación agentes .....	14
Figura 10. Árbol de comportamientos conquista de base .....	15
Figura 11. Estado inicial de la simulación .....	17
Figura 12. Comienza el reclutamiento de los equipos .....	17
Figura 13. Dos equipos van a por la misma base.....	18
Figura 14. Dos equipos luchan por conquistar una base.....	18
Figura 15. Equipo verde conquista la base neutral .....	19
Figura 16. El recluta se ha quedado solo .....	19
Figura 17. Intento de equipo rojo de conquistar la base del equipo verde.....	20
Figura 18. Desaparición del equipo azul.....	20
Figura 19. Dominancia del equipo verde .....	21
Figura 20. Equipo verde intenta conquistar la base morada .....	21
Figura 21. Tan solo sobrevive el equipo rojo a la marea verde .....	22
Figura 22. El equipo verde gana .....	22

## 1. Descripción general

Se trata de una simulación en la que se pueden ajustar los parámetros de forma que se pueden probar cómo varían los distintos comportamientos de los cinco agentes existentes al cambiarlos. Los agentes se tratan de un dron de ataque a distancia, un dron de ataque cuerpo a cuerpo, un reclutador, una torreta y finalmente un entorno que genera estos drones de forma inteligente.

Se ha decidido que el diseño sea 3D con modelos propios desarrollados para esta práctica.

La simulación consiste en una batalla de distintos equipos por conquistar el mayor número de bases en el tiempo de duración de una partida. Esta simulación es escalable, por lo que se puede configurar el número de equipos, cambiar el escenario, la cantidad de agentes y saldrán resultados diferentes. En nuestro caso, se ha decidido crear un total de 3 simulaciones distintas para ver cómo se adapta perfectamente.

El mundo está dominado por distintos equipos diferenciados por su color. Las bases consisten en una serie de nodos repartidos por el escenario que sirven como punto de reaparición para las tropas del equipo al que pertenezcan. En el inicio de una partida todos los equipos están equilibrados con el mismo número de bases y tropas.

Los equipos deberán defender sus bases al mismo tiempo que se coordinan para conquistar bases enemigas. Hay distintos tipos de agentes que interactúan entre ellos para formar estrategias de ataque. A nivel visual, la simulación tendrá una apariencia minimalista ambientada en una sala de experimentación de un laboratorio gigante y los agentes son robots.

Respecto a la implementación del proyecto, no se ha utilizado la librería facilitada debido a que la inteligencia artificial es un campo que interesa a los miembros del equipo y que se pretende estudiar más en profundidad durante el máster. Por este motivo, se ha desarrollado una máquina de estados, un árbol de decisión y un sistema de utilidad propios y genéricos que han servido para el correcto desarrollo de la práctica.

A la hora de ver el proyecto, si se activan los *gizmos*, en la pantalla de *Scene* se puede ver cómo cada dron apunta con una línea su base destino en caso del reclutador o su enemigo a disparar en caso de los drones de ataque.

## **2. Especificación de los personajes**

### **Drones**

Tanto los drones de ataque a distancia, como los de ataque cuerpo a cuerpo como los reclutadores, tienen una clase padre de la que heredan.

Esta clase tiene unos atributos modificables que afectan a los drones que heredan de esta:

- Vida máxima: vida que tiene el dron correspondiente.
- Velocidad de movimiento: velocidad del dron correspondiente.

### **Drones de ataque cuerpo a cuerpo y a distancia**

#### **a. Nombre y descripción textual detallada**

Son los soldados encargados de combatir contra los enemigos, defender y conquistar bases. Su comportamiento está basado en un sistema de utilidad de forma que dependiendo de la situación consideren más importante una acción u otra. Estos drones pueden ser reclutados y seguir las órdenes del reclutador, aunque podrían dar prioridad a otra acción si su influencia es muy fuerte como atacar a enemigos cercanos o defender una base aliada que esté siendo conquistada muy cerca. Su sistema de utilidad es jerárquico, dando lugar cada una de las acciones a una máquina de estados que define su comportamiento.

El estado inicial por defecto es el de aproximamiento de la FSM de ataque. Este estado se ha programado de forma que, en un principio, los drones se mueven de forma aleatoria, haciendo que todas las simulaciones sean diferentes, aunque los parámetros no cambien.

- Tipos de dron: hay dos tipos de drones diferenciados por su velocidad, vida y forma de atacar:
  - a. Cuerpo a cuerpo: son más lentos que los drones a distancia, pero tienen más vida. Su forma de atacar consiste en acercarse al enemigo más cercano y golpearle hasta debilitarlo.
  - b. A distancia: son más rápidos, pero poseen menos puntos de vida. Atacan a distancia disparando, para lo que previamente tendrán que apuntar a su enemigo.

**b. Tabla de percepciones**

<b>Nombre</b>	<b>Implementación</b>	<b>Acceso</b>
Distancia a enemigo más cercano	Calcular la distancia del enemigo más cercano	<i>Pull</i>
Reclutador esperando	Si ha sido reclutado, escuchar evento del reclutador de movilizarse al punto de espera	<i>Push</i>
Reclutador ha dado la orden de conquista	Escuchar evento del reclutador de orden de ataque	<i>Push</i>
Base aliada cercana está siendo conquistada	Comprobar la distancia a la base neutral más cercana que esté siendo conquistada por los enemigos	<i>Pull</i>
Vida agotada	Comprobar si el valor de vida ha llegado a cero	<i>Pull</i>
Ha reaparecido	Escuchar evento del mundo de que ha reaparecido en alguna base	<i>Push</i>
Ha llegado al punto de espera	Comprobar si está en el rango de espera especificado por su reclutador	<i>Pull</i>
Ha llegado al interior de la base objetivo	Comprobar si está dentro de la base que tiene como objetivo	<i>Pull</i>
Enemigo en rango de ataque	Comprobar si un enemigo está en el rango de ataque	<i>Pull</i>
Ha disparado	Comprobar si ha generado un proyectil en dirección al objetivo	<i>Pull</i>
Enemigo en punto de mira	Comprobar si está apuntando a un enemigo	<i>Pull</i>
Ha atacado	Comprobar si ha terminado de atacar cuerpo a cuerpo a un enemigo	<i>Pull</i>

**c. Tabla de acciones**

<b>Nombre</b>	<b>Implementación</b>	<b>Efecto</b>
Atacar al enemigo más cercano	El dron calcula el enemigo situado a menor distancia y a través de una FSM le ataca	El dron ataca a un enemigo
Morir y reaparecer	Al agotar sus vidas la posición del dron cambiará a la de una base aliada, en caso de no existir base aliada, muere para siempre	El dron desaparece y vuelve a generarse en una base aliada
Acercarse al enemigo más cercano	Si el dron es de tipo cuerpo a cuerpo, al detectar un enemigo en su rango de ataque y mediante un algoritmo de <i>pathfinding</i> , se aproxima a este	El dron se mueve hacia el enemigo
Atacar	Si el dron es de tipo cuerpo quita al enemigo cierta vida	El dron genera la animación de atacar cuerpo a cuerpo
Ir al punto de espera del reclutador	El dron a través de una FSM se acerca a la zona establecida y posteriormente espera a las órdenes del reclutador	El dron se dirige hacia el punto de espera y se queda quieto
Acercarse al punto de espera	El dron mediante un algoritmo de <i>pathfinding</i> se dirige hacia el punto de espera establecido por el reclutador	El dron se mueve hacia el reclutador
Esperar órdenes	El dron se queda esperando a que le den alguna orden	El dron se queda quieto
Conquistar base enemiga	El dron a través de un FSM se dirige hacia una base enemiga y la conquista	Nueva base conquistada para el equipo del dron
Entrar en el interior de la base objetivo	Mediante un algoritmo de <i>pathfinding</i> el enemigo se dirige al centro de la base a conquistar	El enemigo entra en la base a conquistar
Defender base aliada	Mediante un FSM el dron entrará en la base que está siendo atacada y atacará a los enemigos que estén intentando conquistarla	El dron defiende su base

#### d. Diagrama descriptivo: Sistema de utilidad jerárquico

Sistema de utilidad Nivel 1:

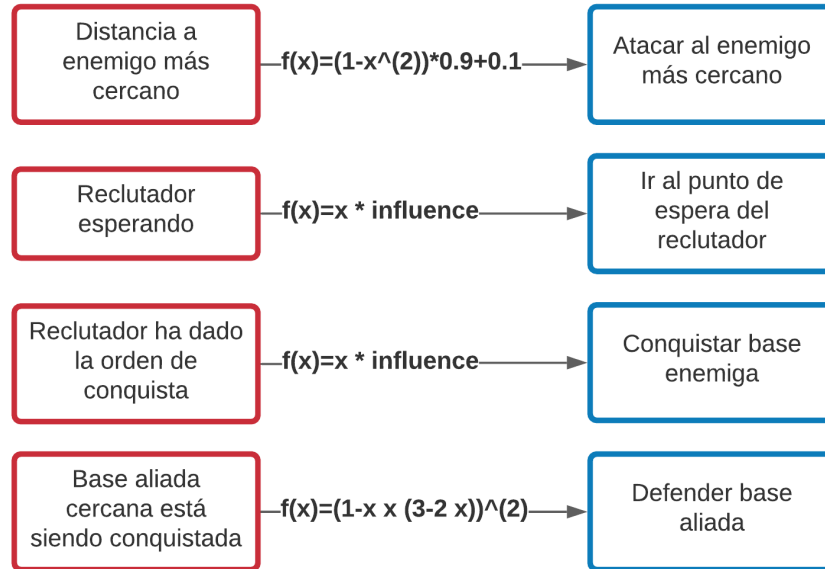


Figura 1. Sistema de utilidad de dron

La primera gráfica refleja la función de utilidad de la rama de Atacar al enemigo más cercano, la segunda las ramas de Ir al punto de espera del reclutador y Conquistar base enemiga, y la tercera la de Defender base aliada.

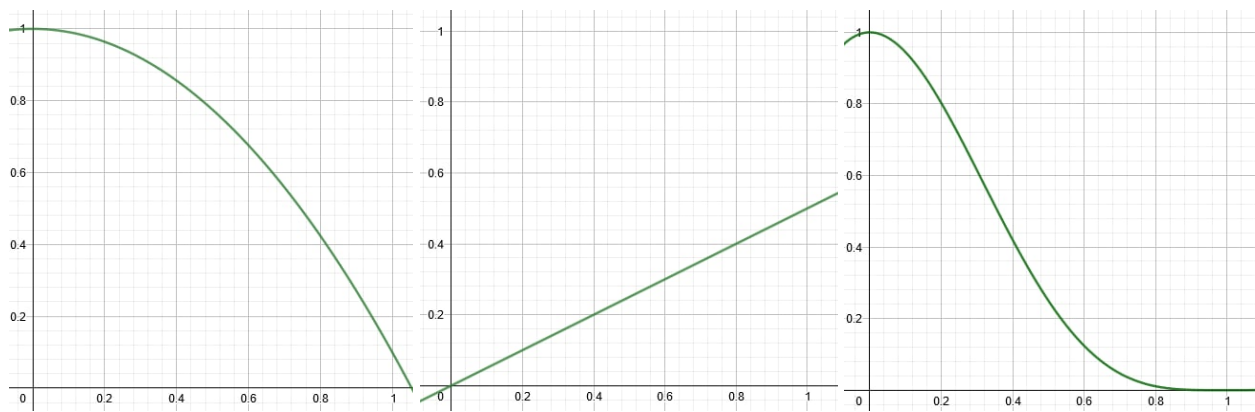


Figura 2 Gráficas de las funciones de utilidad

En el caso de Atacar al enemigo más cercano, se desea que funcione de forma que cuando la percepción tiene un valor menor, es decir, el enemigo está muy cerca, su valor sea muy alto y que este valor decrezca poco a poco al alejarse de un enemigo.



Respecto al de Ir al punto de espera del reclutador y Conquistar base enemiga, las percepciones pueden tener valor de 0 en caso de que no haya recibido orden de ir al punto de espera o no deba conquistar una base enemiga, o 1 en caso de que estas órdenes existan. Es decir, sus percepciones son de tipo umbral. Siendo así, sus funciones de utilidad multiplican estos valores de 0 o 1 por la influencia de estas acciones, de forma que en caso de que la orden sea positiva, el valor de utilidad irá determinado por la influencia del reclutador.

Por último, en el de Defender base aliada, se pretende que, cuando una base muy cercana al dron está siendo atacada este acuda, teniendo un valor de utilidad muy cercano a 1, pero que sin embargo, al estar muy lejos el valor sea muy cercano a 0 de forma que el valor de utilidad no sea lineal sino que crezca y decrezca suavemente.

FSM Nivel 2 (Ir al punto de espera del reclutador):

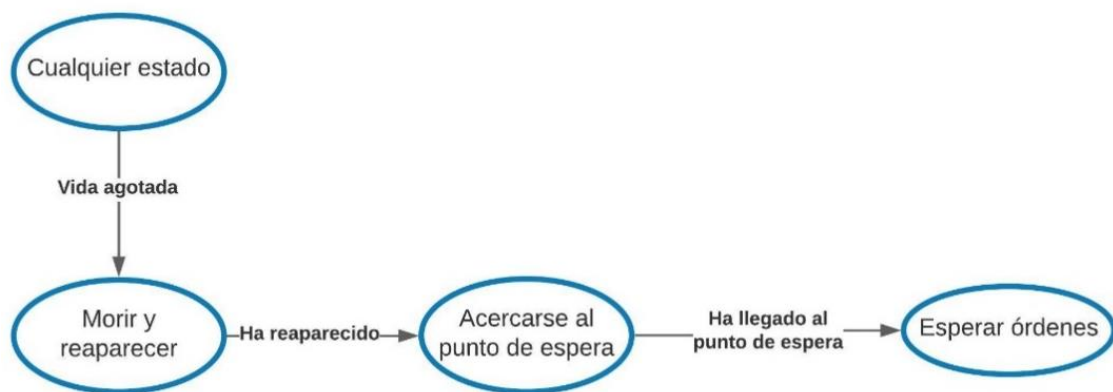


Figura 3. Máquina de estados de dron ir al punto de espera

FSM Nivel 2 (Conquistar o defender base):

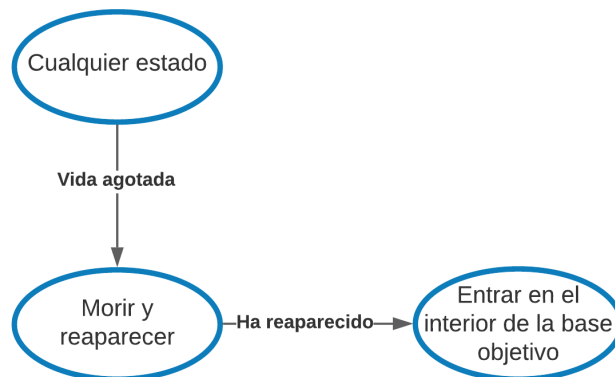


Figura 4. Máquina de estados de dron conquistar o defender una base

FSM Nivel 2 (Atacar enemigo más cercano dron a distancia):

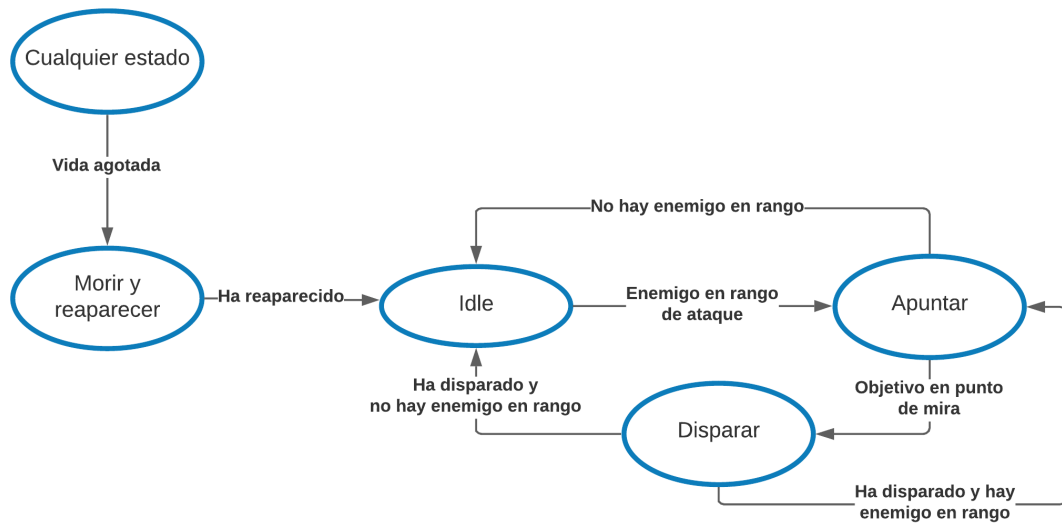


Figura 5. Máquina de estados ataque de dron a distancia

FSM Nivel 2 (Atacar enemigo más cercano dron a cuerpo a cuerpo):

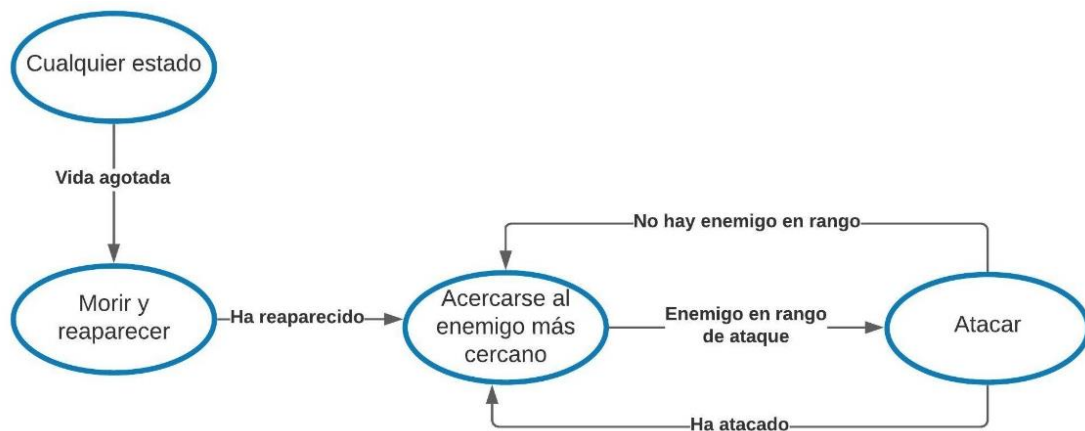


Figura 6. Máquina de estados ataque de dron cuerpo a cuerpo

### e. Estructuras de datos

El dron tiene cuatro atributos modificables generales:

- Rango de detección de enemigos
- Rango de protección
- Daño de ataque

- Influencia del reclutador: se implementará como un cambio en los pesos del sistema de utilidad. Cuanto más alta sea esta influencia menos probable será que el dron de prioridad a acciones que no sean las órdenes de su reclutador.

Además, tiene algunos propios de cada máquina de estados de segundo nivel, que son:

- Respecto a la FSM de ataque a distancia:
  - Cadencia de tiro
  - Velocidad de apuntado
  - Ángulo máximo de tiro
  - Cantidad de movimiento aleatorio al inicio del ataque
- Respecto a la FSM de ataque cuerpo a cuerpo:
  - Rango de ataque
  - Velocidad de ataque
  - Desplazamiento en idle
  - Velocidad de giro

## Reclutador

### a. Nombre y descripción textual detallada

El reclutador es un tipo de dron especial que no se encarga de defender bases o atacar enemigos sino de reclutar aliados. Irá acercándose a sus aliados para reclutarlos en su equipo y cuando haya reclutado suficientes hará un llamamiento a sus reclutas para reunirse en un punto de espera. Este punto de espera es decidido por el reclutador como un lugar cercano a la base no aliada más cercana desde donde no puedan ser atacados por sus torretas. Cuando los reclutas lleguen al punto de espera, el reclutador dará la orden de ataque para intentar conquistar la base. Si no consigue que la cantidad esperada de reclutas lleguen al punto de espera, volverá a reclutar más drones hasta lograr su objetivo. De este agente depende en gran medida la victoria del equipo, pues los drones por sí mismos solo se encargan de defender sus bases por lo que, en el mejor caso, terminarían la partida como empezaron. Sin embargo, este agente permite al equipo ganar territorio a sus enemigos durante la partida.

### b. Tabla de percepciones

Nombre	Implementación	Acceso
Aliado en rango de reclutamiento	Comprobar si el aliado se encuentra dentro del rango de reclutamiento	<i>Pull</i>
Ha reclutado suficientes tropas	Comprobar si el número de tropas reclutadas es suficiente	<i>Pull</i>
Hay aliados suficientes para atacar la base	Comprobar si hay aliados suficientes para realizar una conquista	<i>Pull</i>
Base conquistada	Escuchar evento de la base cuando ha sido conquistada	<i>Push</i>
Vida agotada	Comprueba si el valor de vida es igual a 0	<i>Pull</i>
Ha reaparecido	Escuchar evento de que ha sido regenerado en el mundo	<i>Push</i>

### c. Tabla de acciones

Nombre	Implementación	Efecto
Acercarse al aliado más cercano	El reclutador determinara mediante <i>pathfinding</i> de Unity la posición del aliado y el camino a seguir	El reclutador se acerca a un aliado

Reclutar aliado	El reclutador añade a su lista de tropas influenciadas el aliado reclutado	El aliado seguirá las órdenes del reclutador
Ir a la base enemiga más cercana	El reclutador calculará entre las bases enemigas cual atacar y enviará una orden a las tropas reclutadas para que le sigan.	El reclutador avanza con sus tropas a una base por conquistar
Esperar a las tropas reclutadas	El reclutador espera a que lleguen las tropas reclutadas antes de atacar	Las tropas se colocan en formación para el ataque
Atacar la base enemiga	El reclutador manda la orden de ataque y atacan	Las tropas comienzan su conquista
Morir y reaparecer	El reclutador pierde todos sus puntos de vida y desaparece del mundo	El reclutador muere

#### d. Diagrama descriptivo: Máquina de estados de un nivel

FSM:

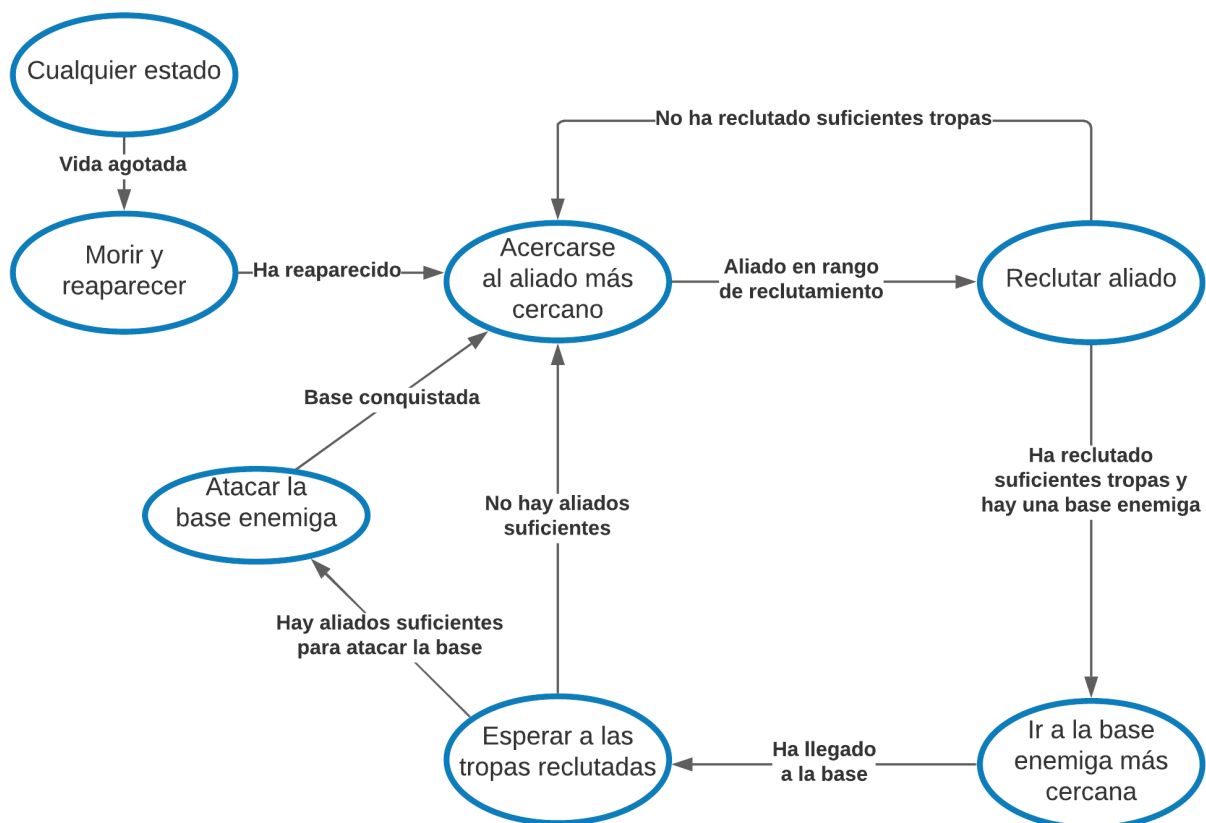


Figura 7. Máquina de estados recluta

#### **e. Estructuras de datos**

El reclutador tendrá cinco atributos modificables:

- Número máximo de aliados
- Rango del punto de espera: distancia máxima respecto a su posición para detectar otros drones están esperando con él.
- Rango de reclutamiento: distancia máxima a la que puede reclutar a un dron
- Porcentaje mínimo de aliados en punto de espera para iniciar un ataque
- Tiempo límite de espera en el punto de espera a los aliados
- Distancia de seguridad: Distancia respecto a la distancia establecida de espera a las bases, la distancia predeterminada viene dada por la distancia de la base a la torreta mas el rango de ataque de la torreta, pretendiendo que el reclutador no pueda ser disparado mientras espera.

## Torreta

### a. Nombre y descripción textual detallada

Las torretas están posicionadas en las bases y se encargan de atacar a todos los agentes no aliados de la base. Por tanto, si la base es neutral, atacarán a todos los agentes. Tienen la limitación de que no pueden atacar al interior de la base, por lo que deben intentar acabar con los enemigos antes de que logren entrar.

### b. Tabla de percepciones

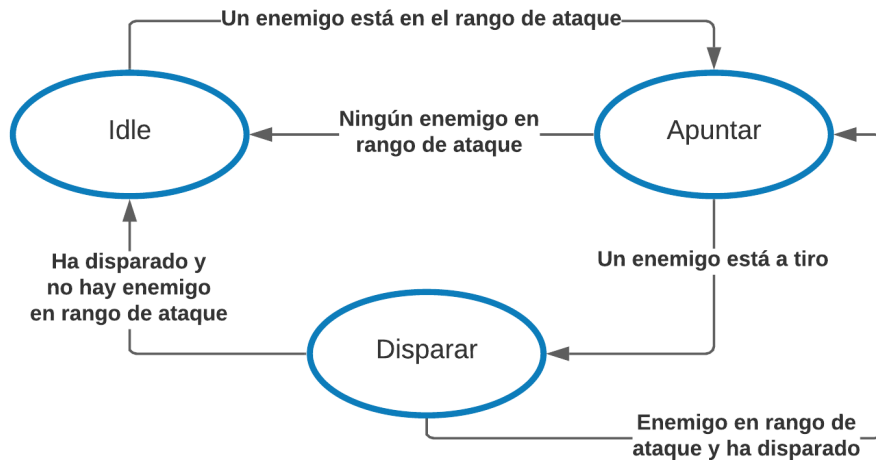
Nombre	Implementación	Acceso
Un enemigo está en rango de ataque	Comprobar si el enemigo se encuentra dentro de su rango de ataque	<i>Pull</i>
Un enemigo está a tiro	Comprobar si se ha calculado la posición aproximada del enemigo	<i>Pull</i>
Ha disparado	Comprobar si se ha disparado al enemigo	<i>Pull</i>

### c. Tabla de acciones

Nombre	Implementación	Efecto
Idle	La torreta espera a que se acerque algún enemigo a la base	La torreta está quieta
Apuntar	Calcula la posición aproximada donde se va a desplazar el jugador	La torreta apunta
Disparar	Dispara una bala que si colisiona con el jugador supondrá una disminución en la energía del jugador	La torreta dispara

#### d. Diagrama descriptivo: Máquina de estados de un nivel

FSM:



*Figura 8. Máquina de estados torreta*

#### e. Estructuras de datos

La torreta tendrá cuatro atributos modificables:

- Velocidad de apuntado
- Cadencia de tiro
- Rango de ataque
- Ángulo de ataque
- Ángulo de disparo
- Daño de las balas



### 3. Interacción con el entorno/mundo

El mundo está dividido en bases de distintos equipos o neutrales. Estas bases consisten en un rango de acción circular protegido por torretas en su perímetro. Las bases sirven a su equipo como punto de reaparición para las tropas, sin embargo, están limitadas a regenerar una tropa cada cierto tiempo. De esta forma, si el equipo pierde muchos drones en poco tiempo, y no tiene las suficientes bases para asumir esa pérdida, no podrá regenerar sus tropas rápidamente y aumentarán sus probabilidades de perder la batalla. El comportamiento de las bases para regenerar enemigos está definido como la siguiente máquina de estados:

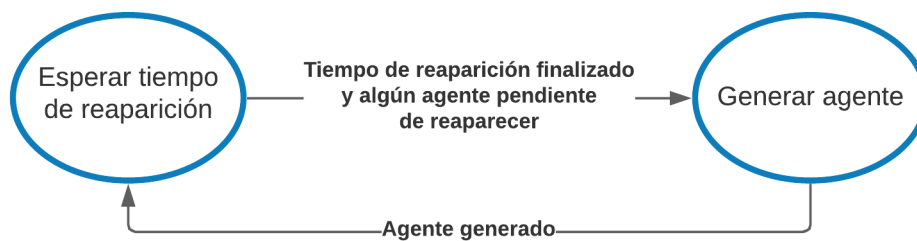


Figura 9. Máquina de estados generación agentes

El generador de agentes tendrá tres atributos modificables:

- Tiempo de reaparición de agentes

Además, existe un *Team Manager* que tiene los siguientes atributos modificables:

- Número de drones cuerpo a cuerpo por equipo
- Número de drones a distancia por equipo
- Número de drones reclutadores por equipo
- Número de equipos
- Colores de los equipos

Para que un equipo pueda conquistar una base enemiga, primero debe neutralizarla. Para esto, sus tropas deberán pasar un tiempo dentro de la base enemiga siendo mayoría respecto a sus adversarios. Una vez la base es neutralizada, del mismo modo estas tropas deberán resistir en mayoría un tiempo dentro de la base para conquistarla.

El comportamiento de las bases para gestionar el equipo al que pertenecen está definido como un árbol de comportamiento de la siguiente forma:

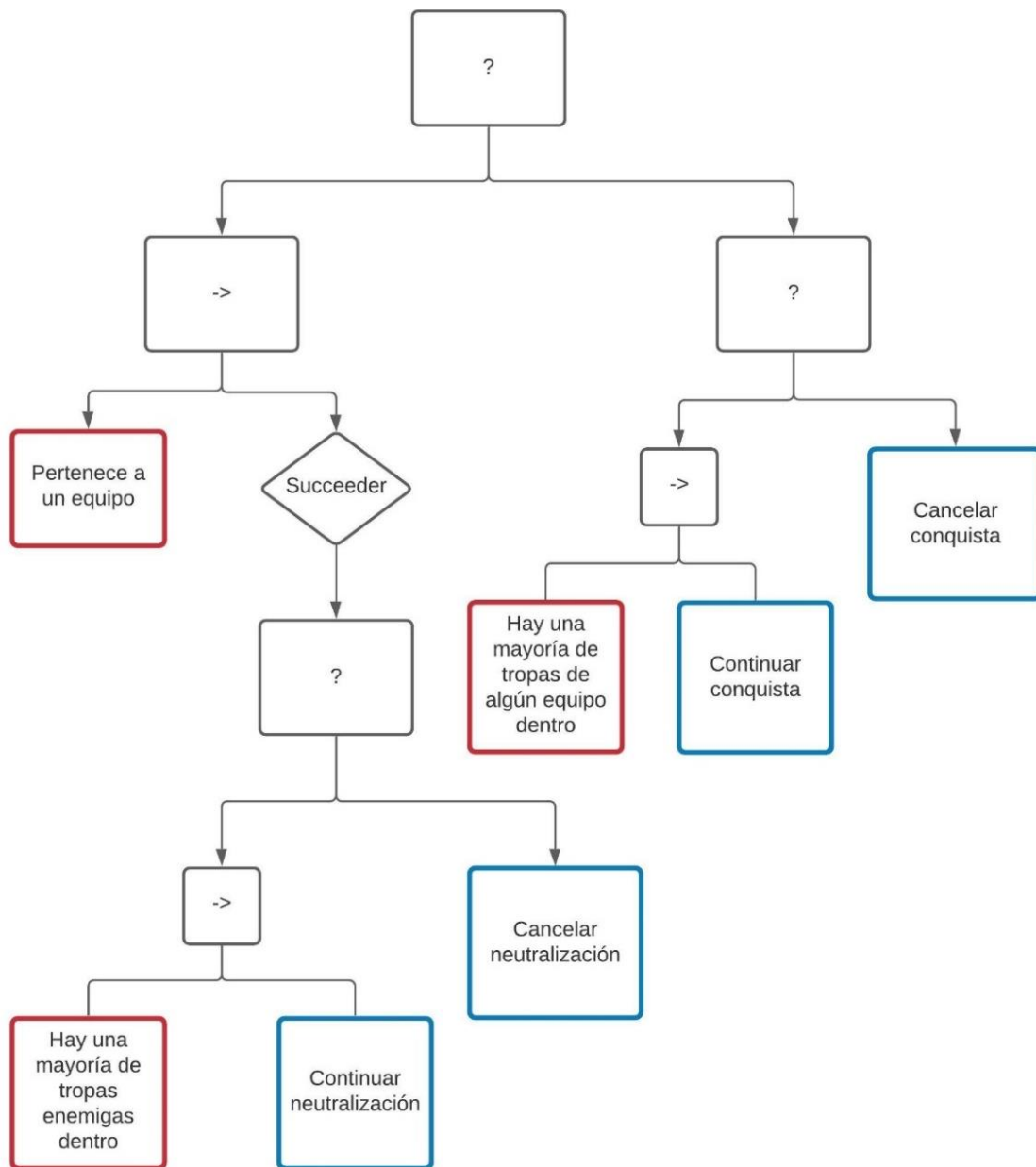


Figura 10. Árbol de comportamientos conquista de base

Las bases tendrán tres atributos modificables:

- Tiempo de neutralización
- Tiempo de conquista
- Equipo al que pertenece

## 4. Resultados

En conclusión, hemos logrado crear una simulación de guerra por equipos basada en la conquista de bases en la que los equipos actúan de forma coordinada e inteligente. Sin embargo, ha habido ciertos aspectos del resultado final que no nos han terminado de convencer. Por ejemplo, en algunas ocasiones el reclutador no sabe qué hacer porque no hay tropas suficientes que reclutar. El tiempo que requieren los equipos para conquistar una base tampoco es algo que nos convenza ya que hace que la simulación sea un poco lenta.

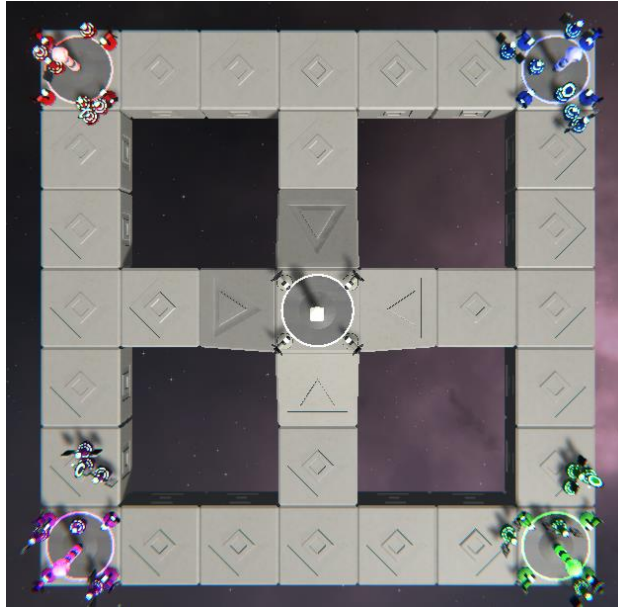
Por otro lado, gracias a los distintos tipos de agentes que tienen los equipos son capaces de complementarse. En especial, queremos destacar la presencia del dron reclutador que se encarga de organizar a los distintos agentes para que los ataques a bases enemigas se realicen de forma inteligente y con una cantidad de tropas adecuada. Además de todo esto, la simulación es completamente escalable y los escenarios modulares. Esto permite crear distintos niveles con la cantidad deseada de equipos muy fácilmente y ver qué tan bueno es el desempeño de los distintos equipos a gran escala.

En nuestro caso, hemos preferido desarrollar los algoritmos para los distintos comportamientos nosotros mismos. Gracias a esto, hemos profundizado y aprendido mucho del lenguaje de programación C# y de cómo hacer el código modular y escalable haciendo uso de distintas técnicas como el uso de *callbacks* o eventos.

Hemos desarrollado los tres tipos de comportamientos explicados en clase: árboles de decisión, sistemas de utilidad y máquinas de estados de forma que las mismas clases se reutilizan entre ellas, haciendo el código mucho más legible y escalable en caso de que más adelante decidiéramos incluir más tipos de comportamiento. Además, nuestra implementación es completamente genérica por lo que es reusable para futuros proyectos sin cambios.

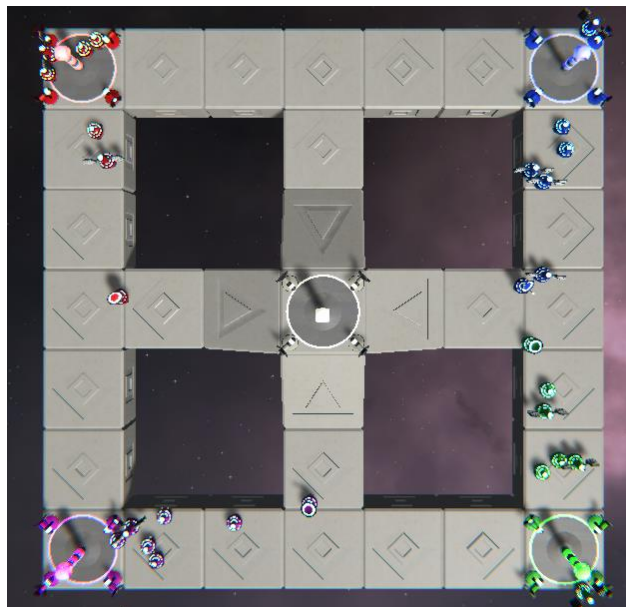
La simulación es muy parametrizable, siendo posible cambiar prácticamente todas las características de agentes y bases. Si bien en un principio pensábamos que la parametrización de los comportamientos no iba a afectar demasiado al resultado final, lo cierto es que en el lapso de un día conseguimos mejorar mucho el desempeño de los agentes cambiando únicamente sus parámetros. Gracias a esto, hemos entendido la importancia de dedicar tiempo a testear la simulación y ajustar cada parámetro hasta lograr el comportamiento deseado.

Por último, nos gustaría destacar también el apartado visual y sonoro ya que no era un requisito para la práctica, pero nos parece que le da un toque más personal y hace que el conjunto se vea más interesante.



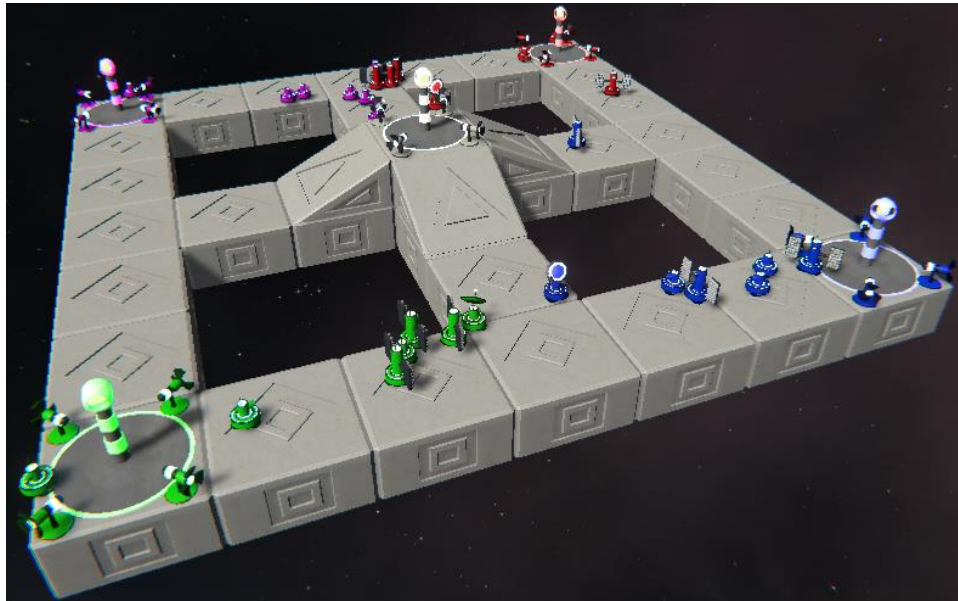
*Figura 11. Estado inicial de la simulación*

Al inicio de una simulación, los drones defenderán sus bases mientras los reclutadores van reclutándolos. Esta es una situación en la que todos los equipos están en equilibrio y sin luchar entre ellos, esperando hasta que el reclutador organice una avanzadilla.



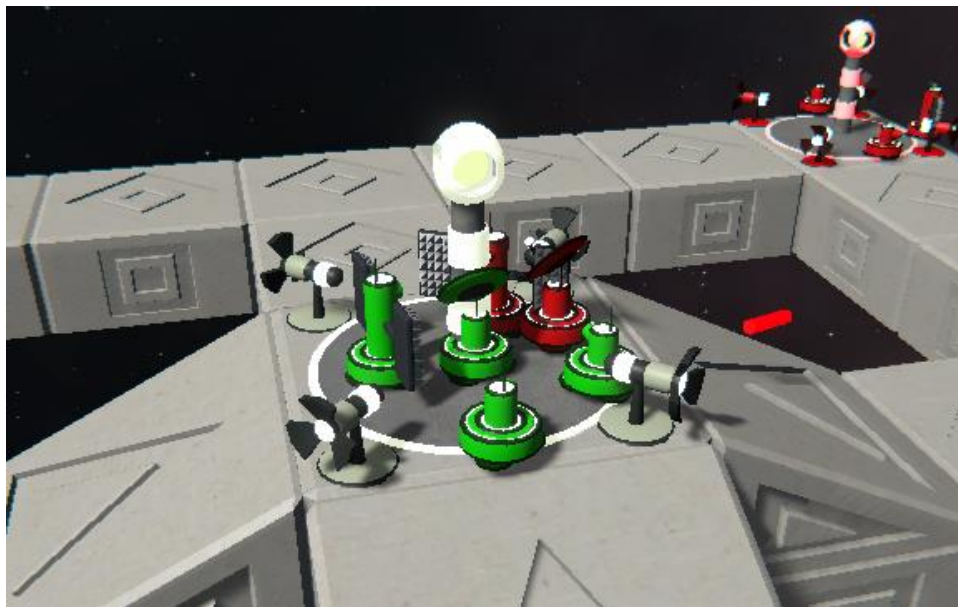
*Figura 12. Comienza el reclutamiento de los equipos*

Una vez los reclutadores comienzan a desplazarse hacia una base para conquistarla, los drones de los distintos equipos comenzarán a atacarlos.



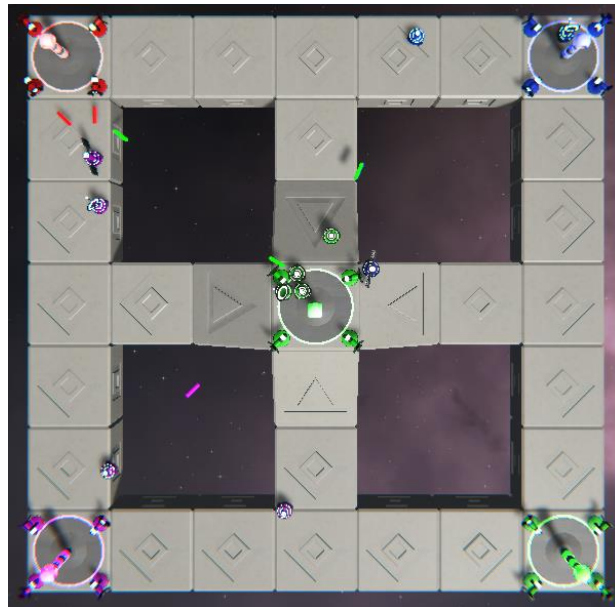
*Figura 13. Dos equipos van a por la misma base*

En ocasiones, más de un equipo intentará conquistar la misma base de forma simultánea, si el punto de espera para realizar el ataque es el mismo para más de un equipo, comenzarán a atacarse antes de iniciar la conquista. Si el reclutador de alguno de los equipos logra sobrevivir y tiene suficientes tropas, este será el que realice el ataque final a la base para conquistarla.



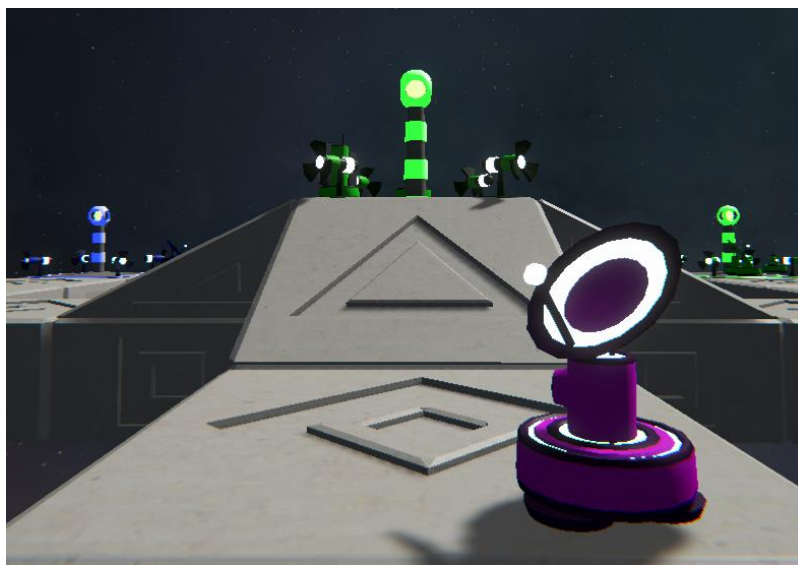
*Figura 14. Dos equipos luchan por conquistar una base*

Si dos equipos intentan conquistar una base al mismo tiempo, tendrá lugar una lucha entre ambos en el interior de la base. El equipo cuyas tropas logren sobrevivir conquistará la base. Los drones de un equipo deberán resistir unos segundos en el interior de la base hasta lograr conquistarla.



*Figura 15. Equipo verde conquista la base neutral*

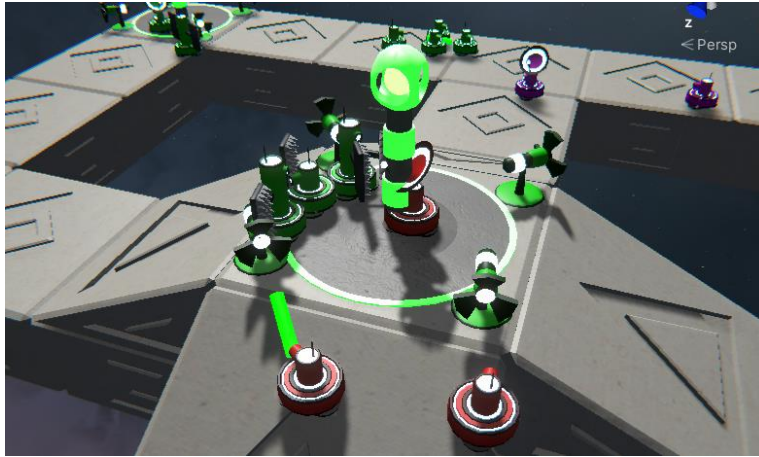
Cuando un equipo conquista bases, gana poder ya que sus tropas reaparecerán más rápido al tener un punto de regeneración extra en el escenario. Esto no solo les da poder sino una posición estratégica desde la cual están más próximos a futuras bases que deban conquistar.



*Figura 16. El recluta se ha quedado solo*

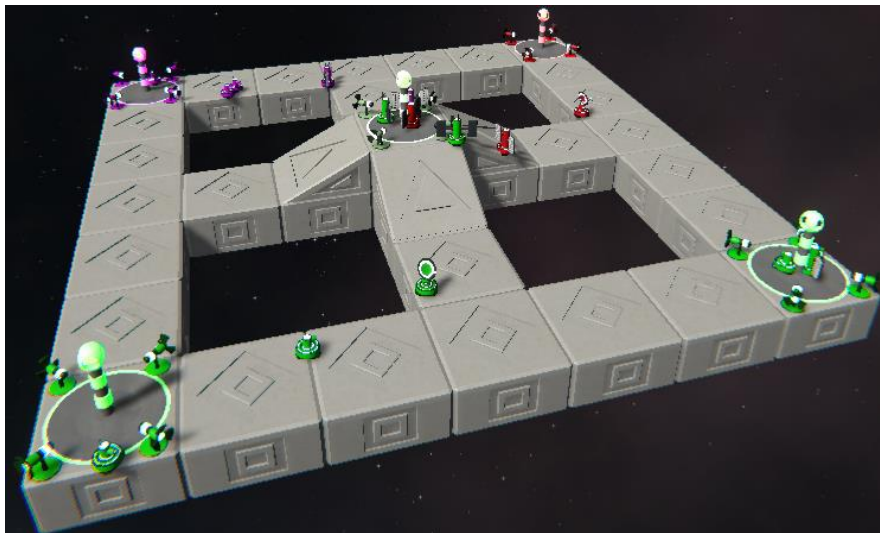


Es posible que, tras haber reclutado suficientes tropas, el reclutador deba cancelar un ataque ya que sus reclutas podrían haber dado prioridad a enfrentarse a enemigos muy cercanos o defender una base cercana que esté siendo atacada. La prioridad que dan los reclutas a las órdenes del reclutador depende del parámetro *recruiterInfluence* que puede ser modificado desde el editor de *Unity* en los *prefabs* de los drones. Este parámetro puede cambiar drásticamente cómo es el desarrollo de la simulación.



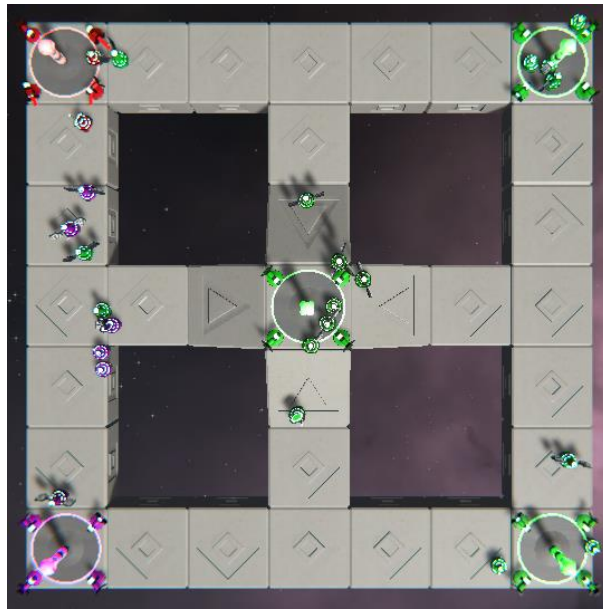
*Figura 17. Intento de equipo rojo de conquistar la base del equipo verde*

Las bases que pertenecen a un equipo son más difíciles de conquistar, ya que son capaces de regenerar tropas en su interior y las torretas no atacan a su propio equipo. Sin embargo, son la única forma de ganar terreno a los equipos enemigos si no hay bases neutrales.



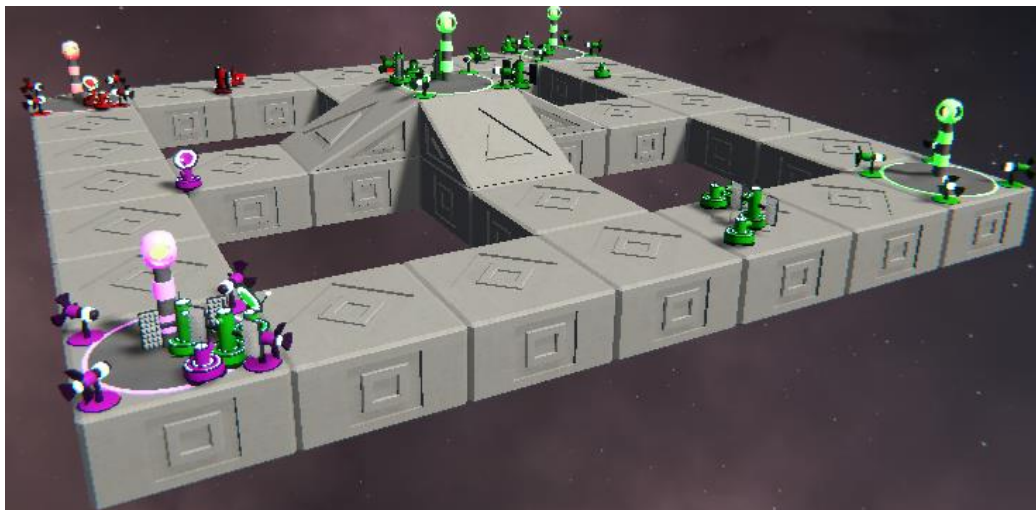
*Figura 18. Desaparición del equipo azul*

En caso de que un equipo pierda todas sus bases, no podrá regenerar tropas. Llegado a esta situación, su única posibilidad será conquistar alguna base antes de perder a todos sus miembros. Sin embargo, esta posibilidad es muy remota por lo que es la mejor forma de acabar con un equipo. Por ejemplo, en la anterior figura puede verse cómo el equipo verde ha conquistado la base azul por lo que ya no está presente en la simulación.



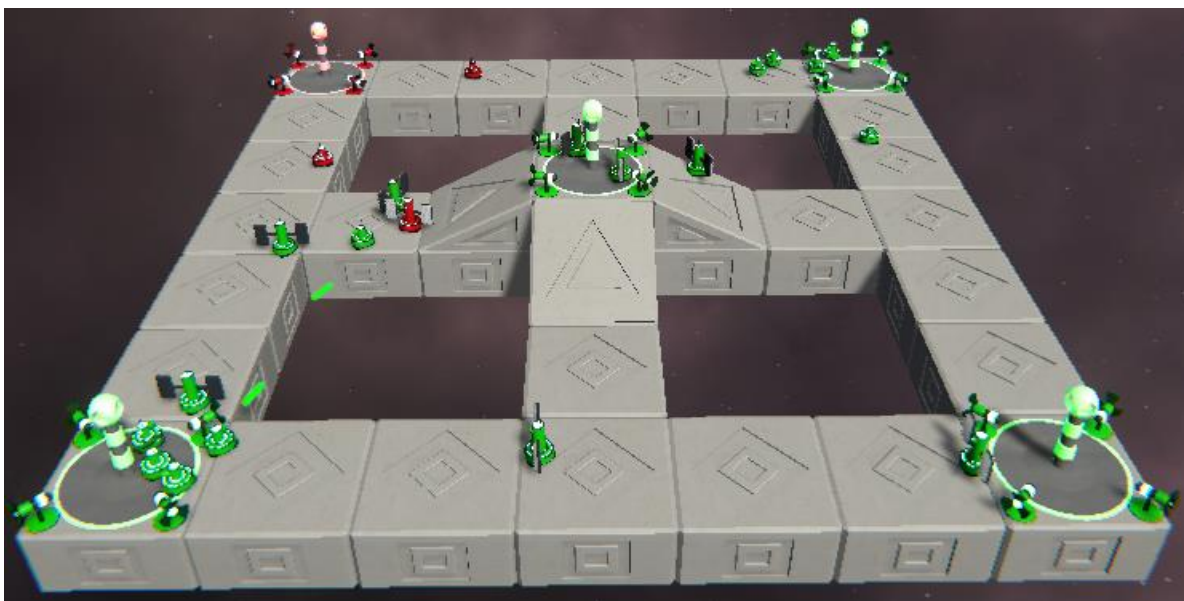
*Figura 19. Dominancia del equipo verde*

Cuando un equipo domina la mayoría de las bases, es poco probable que otros equipos puedan ganarles terreno ya que seguirán combatiendo todos contra todos. Aún así, en ciertas ocasiones si que se puede dar el caso de que un equipo que llevaba ventaja pierda bases y otros le ganen terreno.

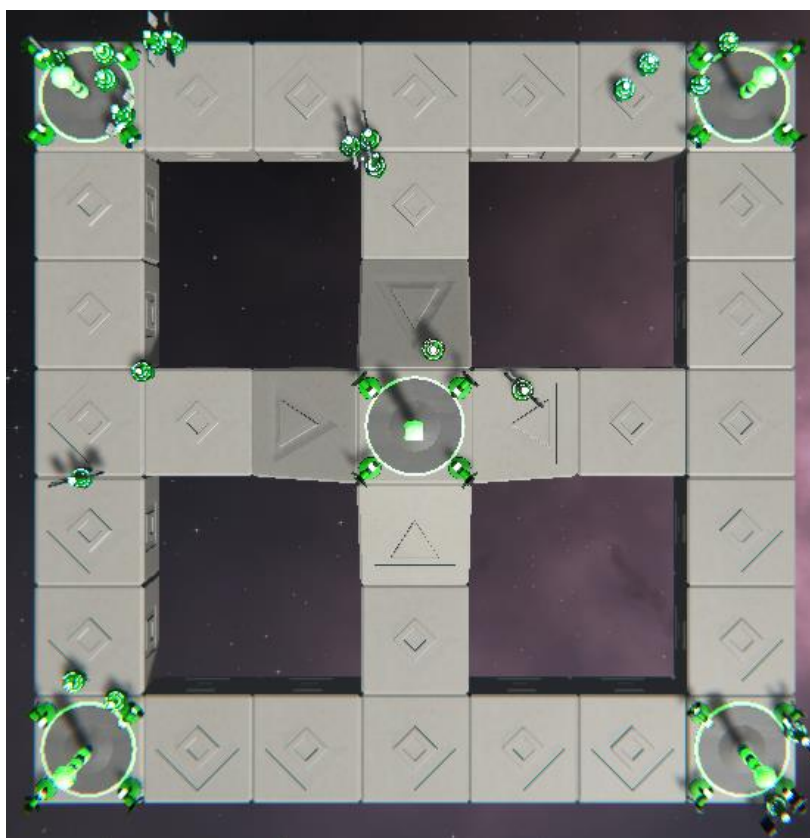


*Figura 20. Equipo verde intenta conquistar la base morada*





*Figura 21. Tan solo sobrevive el equipo rojo a la marea verde*



*Figura 22. El equipo verde gana*

La guerra entre los equipos concluye cuando todas las bases son conquistadas por el mismo equipo y no quedan tropas de ningún otro equipo.

## 5. Reparto de tareas

La tarea de diseño de juego se ha realizado en grupo. El resto de las tareas han sido repartidas entre los tres integrantes del grupo, quedando los roles de la siguiente manera:

- Pedro Casas Martínez: Modelador 3D, programador.
- Juan Manuel Carretero Ávila: Técnico de música y sonido, programador.
- Alejandra Casado Ceballos: Gestora de proyecto, programadora.

Por la parte de modelos 3D, el artista ha realizado los modelos de los agentes y los elementos del escenario para la realización de este proyecto. En cuanto a la música y el sonido, el encargado de ello ha buscado los recursos en internet y los ha editado de forma que se adecuase a los requisitos del proyecto diseñado. Por último, la gestora del proyecto ha redactado la memoria de este y se ha asegurado de que las tareas del resto de miembros fuesen ejecutadas correctamente.

En cuanto a la programación, la división ha sido la siguiente:

- Juan Manuel Carretero Ávila:
  - Creación de la estructura genérica de las máquinas de estados.
  - Programación de la máquina de estados de la generación de agentes.
  - Programación de los sonidos en el juego.
- Alejandra Casado Ceballos:
  - Creación de la estructura genérica del sistema de utilidad.
  - Programación de la máquina de estados del reclutador.
  - Programación del sistema de utilidad de los drones.
  - Programación de la máquina de estados de “Ir al punto de espera del reclutador” de los drones.
- Pedro Casas Martínez:
  - Creación de la estructura genérica del árbol de comportamientos.
  - Programación del árbol de comportamientos del mundo.
  - Programación de la máquina de estados de las torretas.
  - Programación de la máquina de estados de “Conquistar o defender base” de los drones.

- Programación de la máquina de estados de “Atacar enemigo más cercano dron a distancia” de los drones.
- Programación de la máquina de estados de “Atacar enemigo más cercano dron a cuerpo a cuerpo” de los drones.

## 6. Webgrafía

- Brackeys. (14/03/2018). UNITY NAVMESH TUTORIAL – BASICS. Recuperado el 1 de enero de 2021 en el siguiente enlace <https://www.youtube.com/watch?v=CHV1ymlw-P8>
- Garre, C. (fecha de publicación desconocida). TEMA 2 PARTE 1 ARQUITECTURA DE AGENTES – MÁQUINAS DE ESTADOS FINITAS (FSM). Recuperado el 30 de diciembre de 2020 del aula virtual de la Universidad Rey Juan Carlos.
- Garre, C. (fecha de publicación desconocida). TEMA 2 PARTE 2 ÁRBOLES DE COMPORTAMIENTO (BT). Recuperado el 30 de diciembre de 2020 del aula virtual de la Universidad Rey Juan Carlos.
- Garre, C. (fecha de publicación desconocida). TEMA 2 PARTE 3 SISTEMAS DE UTILIDAD (US). Recuperado el 30 de diciembre de 2020 del aula virtual de la Universidad Rey Juan Carlos.
- Bytes, P. (30/06/2017). STARFIELD SKYBOX. Recuperado el 03 de enero de 2021 del siguiente enlace <https://assetstore.unity.com/packages/2d/textures-materials/sky/starfield-skybox-92717>
- Peridactyloptrix. (09/01/2014). SOUND OF AN ANTI-GRAVITY HOVER DEVICE STARTING UP. Recuperado el 31 de diciembre de 2020 de <https://freesound.org/people/peridactyloptrix/sounds/213384/>
- DayCraftMC. (18/02/2016). LARGE LASER BEAM. Recuperado el 1 de enero de 2020 de <https://freesound.org/people/DayCraftMC/sounds/337112/>
- JohnBuhr. (30/10/2015). SWORD CLASHES. Recuperado el 1 de enero del siguiente enlace <https://freesound.org/people/JohnBuhr/sounds/326800/>
- beskhu. (28/03/2013). SIMPLESONG (MAC) CONSONANTS. Recuperado el 31 de diciembre del siguiente enlace <https://freesound.org/people/beskhu/sounds/182153/>
- HenryRichard. (18/11/2018). NES SOUND EFFECTS ROUND 1. Recuperado el 31 de diciembre del siguiente enlace <https://freesound.org/people/HenryRichard/sounds/448266/>
- josepharaoh99. (15/11/2016). ENGINE DYING. Recuperado el 31 de diciembre del siguiente enlace <https://freesound.org/people/josepharaoh99/sounds/368512/>
- Sergenious. (23/06/2008). LASER2. Recuperado el 1 de enero del siguiente enlace <https://freesound.org/people/Sergenious/sounds/55836/>

- Stafeyev, S. (22/10/2020). FREE FLY CAMERA. Recuperado el 5 de enero de 2021 de <https://assetstore.unity.com/packages/tools/camera/free-fly-camera-140739>